# Jurassic News

Retrocomputer Magazine

Anno 5 - Numero 28 - Marzo 2010



**Jurassic News**

Rivista aperiodica di Retro-computing

**Coordinatore editoriale**
Sonicher [Sn]

**Redazione**
Tullio Nicolussi [Tn]
redazione@jurassicnews.com

**Hanno collaborato a questo numero:**
Tullio Nicolussi [Tn]
Salvatore Macomer [Sm]
Sonicher [Sn]
Lorenzo 2 [L2]
Besdelsec [Bs]
Lorenzo Paolini [Lp]
Giovanni [jb72]

**Impaginazione e grafica**
Anna [An]

**Diffusione**
marketing@jurassicnews.com

La rivista viene diffusa in formato PDF via Internet agli utenti registrati sul sito www.jurassicnews.com. la registrazione è gratuita e anonima; si gradisce comunque una registrazione nominativa.

**Contatti**
info@jurassicnews.com



**Jurassic News promuove la libera circolazione delle idee**

# *Marzo 2010*



## *In Copertina*

*Sistema di sviluppo o Personal Computer?*
*L'una e l'altra anima incarnate dalla "piastra" AIM65 della Rockwell, uno dei sistemi che hanno guidato lo sviluppo del Personal Computer già prima che arrivasse il 1980.*

# Editoriale

Ciao mamma, sono contento di essere arrivato uno!

Eccomi dunque qui a dare il mio contribuito organizzativo alla rivista. Come spesso capita (non so a voi, ma al sottoscritto sì), quando affronto una attività che conosco poco e che ha una scadenza molto precisa, vengo preso da una latente preoccupazione di non arrivare in tempo.
Così parto subito in quarta cercando di mettere assieme subito quanto possibile per avere un po' di respiro e avere più tempo per dedicarmi alle cose più sofisticate o delicate da mettere a punto.
Così è stato con JN: non volendo mancare la scadenza del 1^ marzo per la data di pubblicazione, ho cominciato fin da novembre a lavorarci e infatti nelle prime settimane ero molto soddisfatto del punto dove ero arrivato. Poi gli impegni di lavoro e personali hanno avuto il sopravvento al punto che alla fine ci sono arrivato trafelato a confezionare questo fascicolo!

Comunque è andata e devo dire che sono molto soddisfatto del risultato, anche perché gli amici mi hanno consegnato dei "pezzi" veramente al top!
Il risultato è un fascicolo forse fra i più belli che abbiamo pubblicato, così almeno credo io, dall'alto della mia autostima :-)

Un benvenuto caloroso aspetta ad un nuovo collaboratore: Giovanni, che siglerà i suoi pezzi con la firma jb72. Siamo molto felici di accogliere i suoi contributi alla causa comune della cultura retro-informatica!

Sperando che anche voi condividiate il mio entusiasmo, vi auguro buon divertimento nella lettura e una bellissima primavera (è la mia stagione preferita).

***[Sonicher]***

**Jurassic News**

*è una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.*

*La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte (e attentamente vagliate) da Internet.*

# Retrocomputing

## La lunga marcia

*La disponibilità di documentazione e le strategie di incremento e conservazione.*

*Vorrei discutere in questo intervento della disponibilità della documentazione utile all'hobby del retrocomputing.*

*Passando in rassegna la documentazione che riguarda l'hobby del retrocomputing, notiamo esistere una variagata tipologia di documenti che ci sono utili se non addirittura indispensabili.*

*Prima di tutto i manuali hardware e software corredati alle macchine, quando ancora questi avevano un senso; poi le riviste del settore, le monografie specifiche, le fanzine, i manuali a corredo di programmi e i listati stessi, commentati più o meno abbondantemente,... insomma una discreta varietà di oggetti che oggi sono sparsi un po' dappertutto ma che non trovano purtroppo una destinazione definitiva e globale.*

*Grande plauso va fatto a tutte le iniziative che districandosi fra diritti e copyrights riescono a mettere in linea quanto possono. A volte si ha la fortuna di incappare in qualche ripensamento da parte degli editori che danno un provvisorio permesso alla pubblicazione.*

*Gli editori, per quanto male si possa dire di loro arroccati nelle posizioni in difesa di presunti diritti, non hanno proprio tutte le colpe. In realtà la legislatura italiana e internazionale è piuttosto aggrovigliata sul tema di diritti d'autore e chi si azzarda a mettere on-line una qualsiasi cosa presa da un vecchio giornale o uno schema scannerizzato da un manuale, lo fa solitamente a proprio rischio, pronto a cancellare il tutto alla prima lettera minatoria proveniente da presunti detentori di certi diritti che a volte non sappiamo bene decifrare.*

*La reperibilità delle varie tipologie di documenti è diversificata, grazie anche alla diversa sensibilità che ne caratterizza la conservazione da parte dei possesori e alla numerosità che varia da tipologia a tipologia.*

*Ad esempio i manuali a corredo delle macchine: non sono rari, si trovano anche in Internet quasi tutti, vuoi perché in fondo le machine costruite non sono milioni (intendo modelli diversi) e vuoi perché forse il manuale l'acquirente non è stato così stupido da liberarsene in quattro e quattrotto.*

*Possono mancare nella versione italiana o in altro idioma, magari anche significativo, visto che non raramente lo stesso modello veniva venduto in paesi diversi con*

*qualche adattamento: il modulatore TV, tanto per fare un esempio.*

*Le monografie, cioè i libri scritti da autori indipendenti. Se ne sono prodotti a centinaia se non più, più o meno specifici, come ad esempio "Linguaggio macchina per ...", un classico! Questi hanno una vita più lunga perché un volume cartaceo ha una sorta di proprietà virtuale che gli si affibia appena prodotto: la gente è restia a buttarlo. O meglio, esistono moltissime persone che si affezionano ai volumi che hanno comperato, non li prestano e meno che mai li vendono o peggio li bruciano. Trovare questi "tesori", ma anche solo titoli curiosi, è cosa di tutti i giorni se si frequentano le bancarelle dei remainder o i mercatini dei libri usati.*

*Poi ci sono le biblioteche che rappresentano una fonte spesso poco sfruttata dagli appasionati. Soprattutto le biblioteche specialistiche o universitarie sono una vera miniera di tesori da scoprire. Si accede al catalogo on-line, quello che in gergo si chiama OPAC (OPen Access Catalogue) e si cerca ad esempio "Apple". Fra le centinaia di risultati ci saranno i trattati sulla coltivazione delle mele ma anche molti volumi dedicati ai computer della mela. In queste biblioteche, che consiglio senza alcun dubbio, ho trovato delle chicche che nemmeno sospettavo fossero state scritte!*

*Le biblioteche a volte si rivelano anche contenitori di riviste. Qui le cose si fanno meno facili perché la tipologia "magazine" o "periodico", come si dice in Italia, ha degli svantaggi già al momento della produzione. Le riviste spesso vengono buttate perché invecchiano presto e occupano spazio sugli scaffali e questo anche nelle biblioteche pubbliche, non solo in casa propria. Poi c'è il problema della completezza: quante volte abbiamo visto o reperito intere collezioni di periodici alle quali, castigo di Dio, manca qualche numero? Che disdetta! Eh sì, perché il numero mancante "buca" la collezione che non si può dire così "completa". Poi guarda caso è sempre sul numero che ci manca che ci sarebbe stato quel tale articolo, quella tale prova hardware, quella seconda puntata della guida alla quale avremmo tenuto tanto,...*

*La produzione di riviste nel settore computing è stata imponente, soprattutto dall'80 al 2000. Cercando on-line si trova qualche repository ben fornito, altri con pochi sparuti numeri e qualche copertina, altri ancora che promettono la scansione dei testi ma poi non se n'è fatto nulla...*

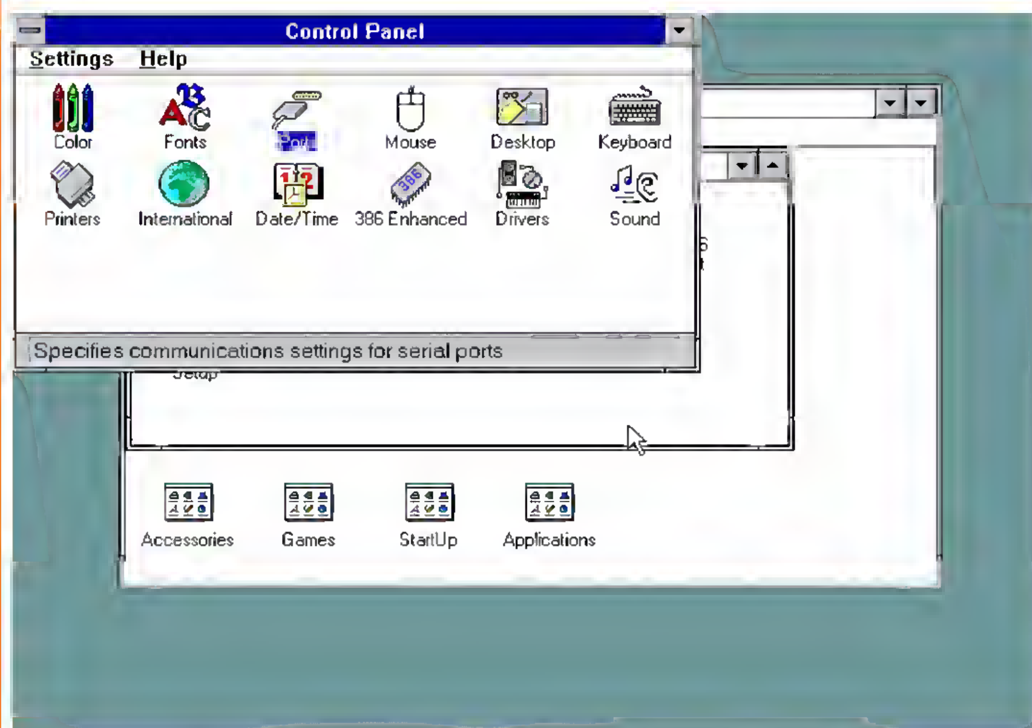
*Da anni mi sto dedicando proprio a questa attività, assieme all'amico Sonicher: recuperare tutto il possibile di ciò che appare in Internet nel settore riviste per PC. E' una strada lunga e forse che non ha una conclusione, ma come diceva il saggio: -"anche un lungo viaggio comincia con un piccolo passo".*

***[Tn]***

# Come eravamo...

La storia dei sistemi e degli uomini che hanno creato un mondo nuovo.

## Installare Windows 3.1




*Vi ricordate i favolosi anni '80? Beh, favolosi per noi che ci occupavamo di informatica: tante cose da scoprire, tante tecnologie annunciate, proposte, sperimentate. Il mondo stava diventando degli informatici, ci sentivamo Dio in Terra e infatti così è stato, cioè il mondo è degli informatici oggi, anche se siamo usati prevalentemente come comprimari, non come protagonisti. Per quanto riguarda la deità, questa è rimasta appannaggio del Dio Denaro.*

*Vi ricordate la prima versione di Windows? Parlo della 1.0, uscita il 20 novembre 1985 e arrivata in Italia qualche mese più tardi, diciamo la primavera del 1986. Chi ha avuto l'ardire di installarla (ammesso ci sia riuscito) sul suo clone IBM con 8088 e 512 Kb di RAM, se non ha vomitato subito probabilmente è stato solo perché s'è sbrigato a cancellare tutto chiedendo perdono al suo fedele, per quanto vituperato, DOS 3.0.*

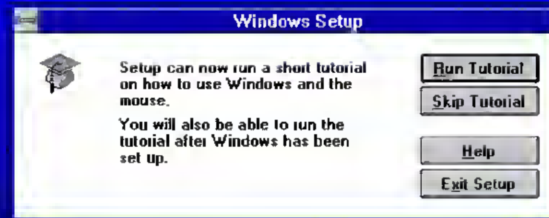
*Dopo questa prima uscita, le altre versioni: la 2.0, poi lo sdoppiamento in /286 e /386 per cercare di sfruttare quel poco di più che i processori Intel di nuova concezione potevano offrire. Le due versioni sono poi confluite nella versione 3.0 del 22 maggio 1990. Ma non ci siamo ancora, bisognerà aspettare la release 3.1, codice segreto*

*"Janus", per avere un qualcosa di decente che riesca anche ad utilizzare il suono (multimedia extension) del 6 aprile 1992.*

*Non sono passati nemmeno vent'anni e probabilmente molti di noi non si ricordano che vagamente di quel primo sistema a finestre, così lento e farraginoso, ma affascinante per noi abituati a combattere con l'interfaccia a caratteri installando tutti i possibili ammenicoli che ci aiutassero a districarci nel labirinto dl file-system che cominciava a crescere a dismisura.*

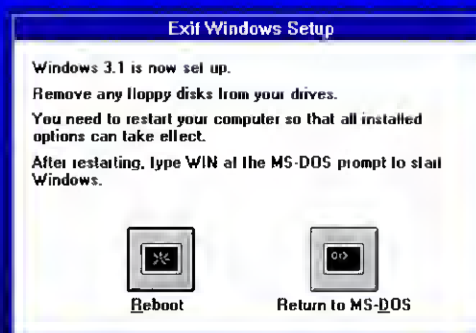
*L'avevamo installato un PC con Windows in azienda, era l'estate del 1992, negli uffici pochissime persone e soprattutto nessun dirigente in giro! Ricordo che lo facemmo in due, il mio collega e io, visto che eravamo gli unici a capirci qualcosa di sistema operativo. I cinque floppy vennero su pian pianino, come l'uso, sul quel catafalco che eravamo riusciti a recuperare pari ad un clone 80386sx, manco a colori che la scheda era una vecchia EGA (con l'Hercules non aveva voluto saperne).*

*Il problema vero fu che mica ce lo avevamo un mouse in ufficio, così che dopo il fatidico WIN <INVIO> ci trovammo imbarazzati a cercare disperatamente le combinazioni di tasti per aprire i menù e spostare le finestre. Andammo a rovistare nel magazzino dove Giuseppe (il collega che se ne occupava) si fece bontà sua "rubare" (dicendo "Io non ho visto niente") un mouse Microsoft ancora imballato con tanto di scheda dedicata. Ma non era come attaccarlo ad una porta USB, bisognava*







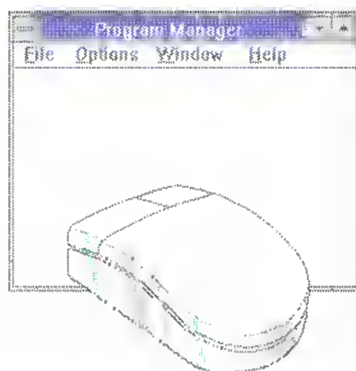


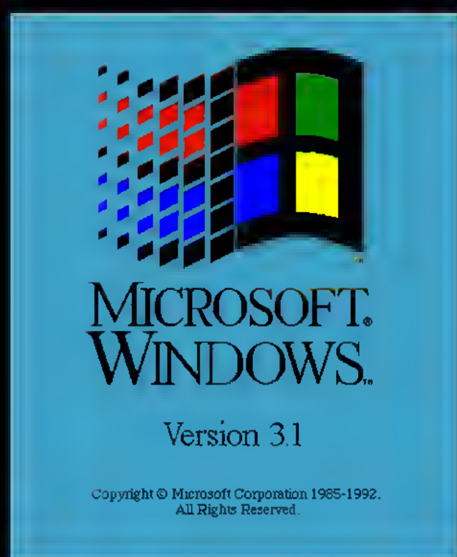
This Tutorial has two lessons.

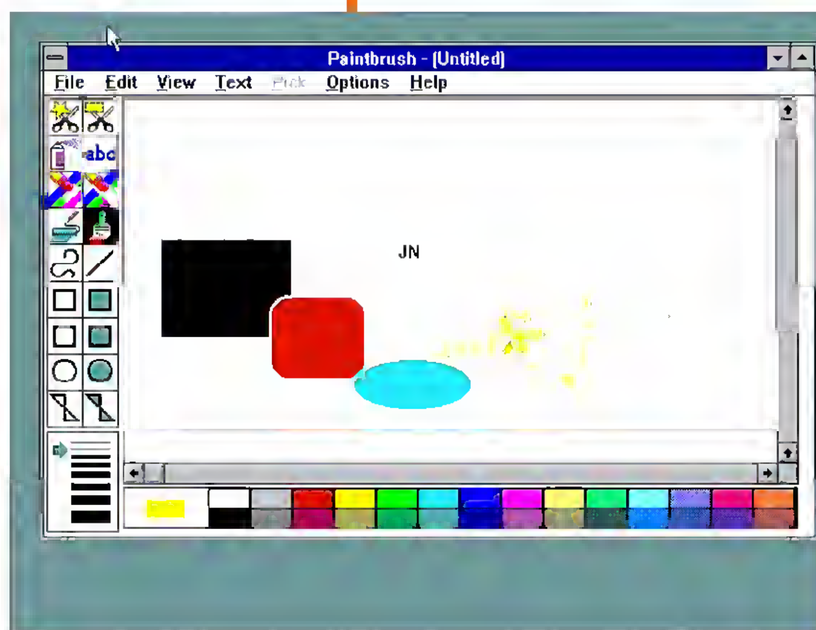
- If you want to learn how to use the mouse, or if you need to brush up on your mouse skills, type **M** to begin the Mouse lesson.
- If you are already a skilled mouse user, type **W** to begin the Windows Basics lesson.

Or, if you want to run the Tutorial at another time:

- Press the **ESC** key to exit the Tutorial.




lavorare di driver DOS, installando il dischetto allegato e poi Windows, bontà sua, si accorgeva che c'era.

Per dirla tutta questo Windows 3.1 ci piacque abbastanza; non è che ne fossimo proprio digiuni avendo visto la versione 2.0 e prima l'improbabile DOS Executive. Come molti utilizzatori "power" del DOS, il nostro preconcetto era che Windows non serviva a nulla se non a perdere tempo e fare dei brutti disegnini, seppur coloratissimi, con il mouse. Cosa serviva, se poi l'unica possibilità era stampare in bianco/nero?

Ne esplorammo però tutti i programmi che lo componevano: il file manager prima di tutto, il pannello di controllo e i giochi. Devo dire che passammo delle ore sul solitario e sul campo minato, però ci vennero ben presto a noia, ma che volete farci, questo passava il convento: o il solitario o lavorare... meglio il solitario!

Che sia stata questa prima sperimentazione o le successive, fatto sta che il mio collega e il sottoscritto furono i primi e per lungo tempo anche i soli, a capirci qualcosa di Windows soprattutto come si programmava. Merito questo del mitico volume di Petzold "Programming in Windows" che studiammo fin dai minimi particolari con il nostro incerto C e il compilatore Microsoft che riuscimmo a farci comprare quell'inverno.

*Che dire, un pizzico di nostalgia, non già per la povertà degli strumenti cui disponevamo, né per la inverecconda indisponibilità aziendale nello spendere due lire per non rimanere indietro con lo sviluppo dei suoi prodotti. Non ci credeva questa nostra azienda, che abbandonammo entrambi pochi anni dopo, nel mouse e nello schermo a colori. "A cosa servono queste stupidaggini? Basta il DOS" ci ripeteva il nostro capo reparto tecnico, responsabile dello sviluppo software. Ma lui, lo si sapeva, non sopportava i PC e di conseguenza la gente che ci lavorava: gli unici sistemi di calcolo seri erano il mainframe e i mini dipartimentali, tutto il resto erano "baracche da sala giochi"... Credo si stia ancora mangiando le mani per non aver capito in tempo che il mondo stava cambiando, anzi che era già cambiato e lui si è ridotto a fare il consulente organizzativo.*

***[Tn]***

# Le prove di Jurassic News

## Rockwell - AIM 65/20

C'è stata una stagione, alla fine degli anni '70, dove la dizione "Personal Computer" non aveva significato e quello che si pensava potesse servire era semplicemente un sistema per imparare ad usare il micro processore.




Una bella immagine prospettica del sistema di sviluppo della Rockwell; AIM 65 o R6500.

### Contesto storico

Siamo nell'anno 1976 e il computer personale non è ancora una idea acquisita. Bisognerà aspettare la geniale mente di Wozniak per avere a breve l'Apple 1. I microprocessori vengono prodotti ma in realtà non si sa che cosa farne. Non che le idee non ci siano, ma questo chip è troppo complicato e le sue potenzialità sono ancora in fase di studio da parte dei progettisti.

In questo scenario hanno assunto estrema importanza i sistemi di sviluppo messi a punto dalle stesse ditte produttrici dei chip con lo scopo dichiarato di favorire la comprensione dell'elettronica digitale supportata dal calcolatore ai nuovi progettisti.

Non è che non esistessero prima i sistemi di sviluppo, i microprocessori e la documentazione, solo che era estremamente complicato usarli. Ad esempio l'8080 di Intel richiedeva una serie di chip di supporto indispensabili a far funzionare un circuito e la "potenza" non è che fosse così elevata. L'altro source, la Motorola, produce il 6800; ottimo chip ma costoso e anch'esso complicato.

Chi produce il 6502, derivato direttamente dal 6800, ha necessità di far conoscere le potenzialità

*del nuovo chip ed ecco che nascono i vari Kim-1 dalla MOS Technology e appunto l'AIM 65 dalla Rockwell International con sede ad Anaheim in California, second source per la produzione del chip.*

*Una persona all'epoca impiegata presso Rockwell, tale Mark Reardon, sostiene che la ditta prevedeva di vendere dai 400 al 600 sistemi ma che alla fine furono più di 50.000 (fonte: old-computers.com Museum ~ Rockwell AIM 65).*

*Il prezzo del computer con 4 Kb di RAM si aggira attorno ai 375 dollari, mentre una versione "Light", senza tastiera e stampante, viene venduta attorno ai 175 dollari.*




*Il sistema visto da sopra*

## *Primo approccio*

*L'AIM 65 è un computer single-board che viene venduto "nudo", cioè senza cabinet (che apparirà qualche tempo dopo) e senza alimentatore. L'utente compra in pratica una piastra elettronica densa di componenti, una tastiera alfanumerica di tipo telescrivente, da collegare alla piastra madre con un cavo piatto di pochi centimentri e una pletora di documentazione.*

*A questo punto ci si deve preoccupare dell'alimentazione, per la verità non tanto difficoltoso come compito e soprattutto di imparare ad utilizzarlo questo sistema, con l'aritmetica esadecimale, le istruzioni mnemoniche, gli indirizzi di memoria, etc...*

*L'output è assicurato attraverso un display a led di un'unica riga da 20 caratteri e da una stampantina termica delle stesse caratteristiche cablata direttamente sulla piastra madre.*

*Rispetto alla concorrenza, l'AIM 65 gode di una maggiore integrazione e professionalità che si tra-*






*In questa pagina una breve rassegna di contenitori approntati per rendere il sistema più simile ad un vero calcolatore, oltre che proteggere l'elettronica da danni accidentali.*









*duce nell'avere una vera tastiera per l'input dei dati, al posto della scomoda tastierina esadecimale dei sistemi di sviluppo concorrenti. Oltre a questo l'ampiezza del display, non limitato ai quattro byte di indirizzo e due byte per i dati, offre maggiori possibilità di editing e di esecuzione del codice. Infine poter ottenere su carta il listing/dump della memoria è ulteriormente gradito.*

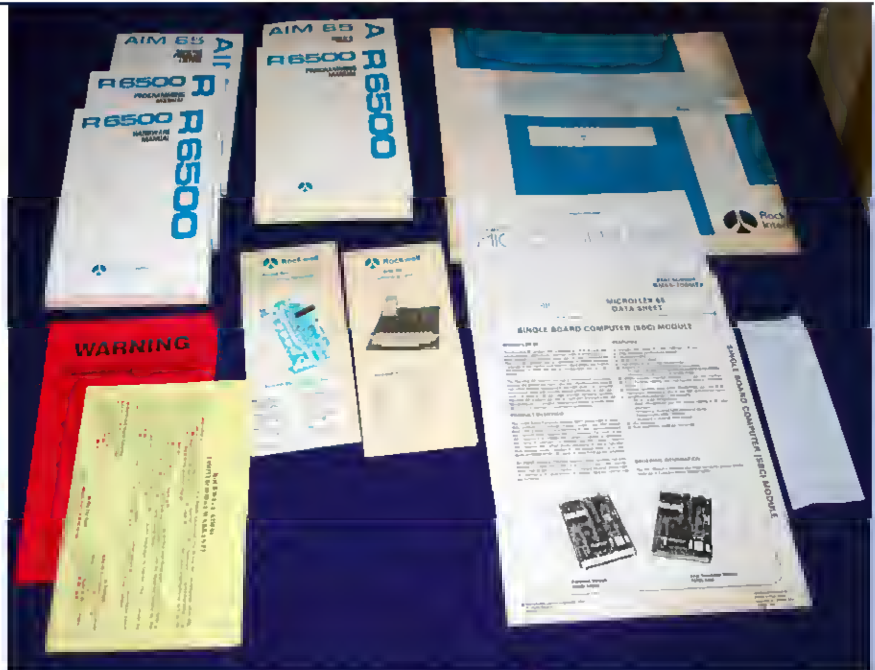
*La seconda cosa di cui preoccuparsi, dopo aver costruito l'alimentatore e scoperto come alimentare il sistema, è quello di trovare un panno morbido ma non troppo e soprattutto non conduttore, per appogiarvici sopra tastiera e piastra madre. Un'altra accortezza è prevedere un cavo di collegamento con la tastiera un attimino più lungo dei miseri sei centimetri di quello originale, permettendo un posizionamento meno rigido delle due unità una dall'altra.*

*A questo punto avrete già letto il manuale e scoperto che i due connettori a pettine che sporgono dal retro della piastra si chiamano "Application" e "Exspansion".*

*La piastra supporta nativamente due registratori audio compreso un segnale per l'azionamento dei motori di trascinamento della cassetta e una telescrivente in current loop da usare al posto della tastiera e della stampante nativa.*

*La qualità della tastiera è ottima, garantendo un feedback alla digitazione non stancante, anche dopo ore di utilizzo. Qualche riserva sul display per via della luminosità non esaltante e della dimensione dei caratteri forse troppo minuscola. La stampante, pur non essendo in tecnologia ad impatto, è comunque abbastanza rumorosa perché utilizza una tecnica di "scarica ad alta tensione" per bruciare la carta e far apparire i caratteri.*

*La carta termina da utilizzare per la stampante, un rotolo è contenuto nella confezione acquistata, è abbastanza reperibile perché usata anche da alcuni modelli di calcolatrici da tavolo e registratori di cassa, ma è costosa e certo non da utilizzare con leggerezza se non per un listing*




*L'abbondanza della documentazione fornita in dotazione.*

*La pagina pubblicitaria più classica dell'importatore italiano, apparsa su tutte le riviste di settore.*

*finale, piuttosto che attivare il semplice eco del video che riporta tutto quanto appare sul display anche su carta.*

*Assieme alla macchina arrivano quattro manuali che sono:*

```
- AIM 65 User’s Guide
- AIM 65 Monitor Program
Listing
- R6500 Microcomputer System
Programming Manual
- R6500 Microcomputer System
Hardware Manual
```

*Inoltre altra documentazione e informativa la troviamo negli allegati:*

```
- R6500 Programming Reference
Card
- AIM 65 Summary Card
- Warranty Card
```

*Dai manuali si capisce che questo prodotto viene chiamato indifferentemente AIM65 o anche “R6500 Microcomputer”, intendendo per la Rockwell essere un vero e proprio sistema di sviluppo per la sua soluzione di micro calcolo ma soprattutto micro-controllo.*

*La lettura di questi manuali è una vera miniera di informazione che può costituire autonomamente un corso pratico di elettronica digitale, anche se una infarinatura è meglio averla già prima…*

## Hardware

*La CPU è un Rockwell mp6502 funzionante a 1 Mhz montato su zoccolo; per le caratteristiche del 6502 questo significa che una istruzione semplice viene eseguita in due microsecondi. Assieme alla CPU la piastra ospita 4 Kbyte di RAM statica (ma ne esiste una versione “Light” che porta a bordo un solo K), e 4 Kbyte di ROM contenente il monitor di sistema. La ROM è espandibile fino a 20 Kb grazie alla presenza di cinque zoccoli in grado di ospitare ROM da 4Kb e che permettono di avere un ottimo sistema comprensivo di editor, assemblatore simbolico e disassembler. La versione venduta in Italia ha già 4 Kb di RAM e monitor espanso a 12 Kb con editor, assembler e disassembler. Esiste anche una ROM BASIC da 8 Kb da usare al posto della ROM assembler e altre con altri linguaggi come il PL/65 (una variante del PL/1), il PASCAL, il FORTH, etc… che richiedono però una maggiore espansione di RAM, ottenibile solo via card esterna, per essere utilizzati con qualche profitto.*

*Le due porte di I/O sono già predisposte per una interfaccia a cassette magnetiche e per un terminale TTY. A questo proposito sulla piastra è presente uno switch che esclude le periferiche interne (display e stampante) in favore di un terminale seriale.*

*Un pulsante di Reset è accessibile sulla piastra in prossimità della tastiera.*

*Come chip di supporto il sistema porta un 6532 per il controllo del timing della memoria, un VIA (Versatile Interface Adapter) 6522 e un PIA 6520 (Peripheral Interface Adapter).*

*La stampante a venti colonne è in tecnologia termica in grado di stampare un carattere formato da una matrice 5x7 con riga di stampa di 20 caratteri (come la riga del display) su un nastro di carta continuo di larghezza 2 pollici e un quarto.*

*Il display a led è costituito da unità a 16 segmenti, il che permette di riprodurre tutti e 64 i caratteri del set ASCII standard.*

*La tastiera è una QWERTY a 54 tasti e comprende il codice ASCII più tasti di controllo (Control, Shift, Del, Escape) e lascia liberi tre tasti (siglati F1, F2 e F3) per la definizione di funzioni utente.*

*L'alimentazione non è fornita con la macchina (ma ne viene offerto uno come opzione se si preferisce non costruirselo), consiste in due tensioni: +5Volt per l'elettronica (2 Ampere) e +24Volt per la stampante termica (2.5 Ampere) ed usabile anche come linea di alimentazione per altre unità collegate attraverso le porte di espansione.*

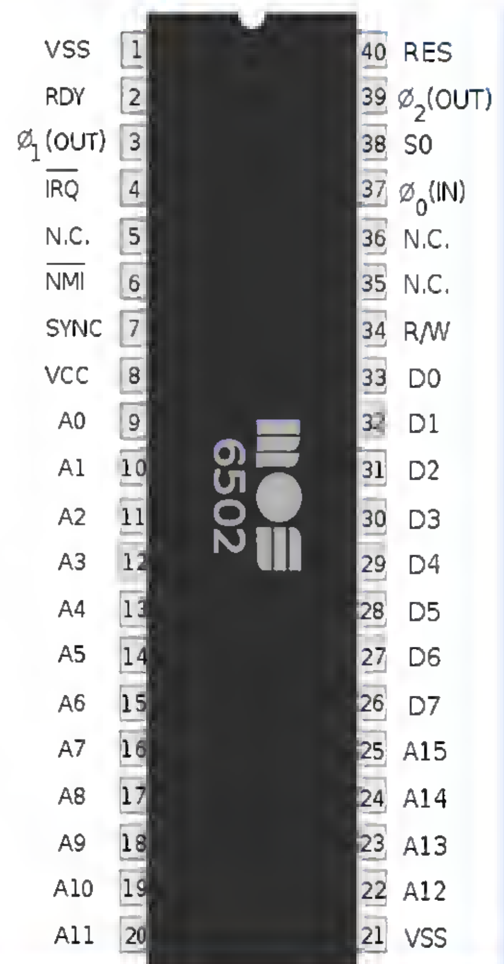
*Il connettore di alimentazione presente sulla piastra, dotato di una morsettiera robusta, riporta anche i connettori per le tensioni +12 e -12 Volt, da usarsi nell'eventualità si usino chip con queste necessità, ad esempio EPROM.*

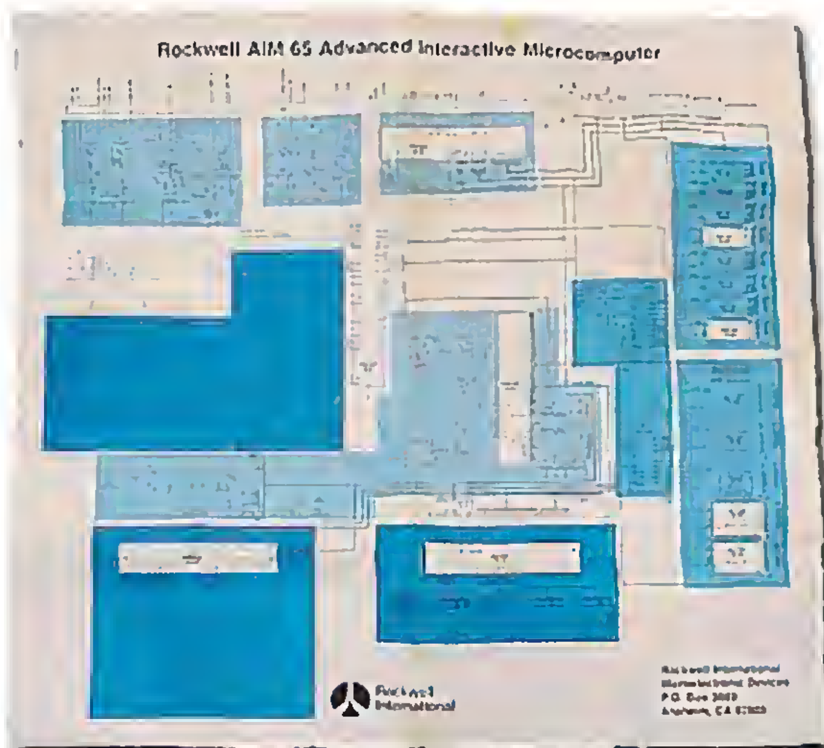
*La struttura del progetto è sufficientemente espansa da consentire l'approntamento di esperimenti di programmazione e controllo anche abbastanza sofisticati, grazie alle due porte da 8 bit programmabili, il contatore digitale e gli slot di espansione che portano all'esterno tutti i segnali dei bus di sistema. L'esposizione dei segnali all'esterno non è protetta da chip di buffering, cosa che sarà prudente attuare a fronte dell'uso con schede di espansione*




*Un particolare della coppia tastiera-display.*

*(Sotto) la piadinatura del micro processore 6502*






*La "mappa" hardware del sistema evidenzia i vari blocchi funzionali per la facile identificazione delle componenti.*

*Il display a 16 segmenti prevede una configurazione dei led come quella riportata in figura.*



*di nostra progettazione.*

*Limitata anche la possibilità nativa di espandere la RAM che richiede all'occorrenza la predisposizione (o l'acquisto) di una apposita scheda in grado di codificare gli indirizzi necessari, oltre a portare a bordo i chip di memoria statica.*

*I due connettori di espansione posti sul retro della piastra assomigliano sorprendentemente, anche nel nome, a quelli del KIM-1, un progetto analogo della MOS Technology. Infatti sui manuali vengono dichiarati compatibili con quelli del KIM, salvo poche e documentate differenze. Probabilmente le due società fornitrici del 6502 si sono accordate su questo punto o più semplicemente il progetto è stato elaborato da un team comune e poi differenziato nei particolari e nel target di riferimento (l'AIM65 costa sensibilmente di più di un KIM-1).*

*Il pilotaggio dei due registratori a casetta che sono collegabili alla macchina prevede dal punto di vista del segnale due pin di uscita con un livello basso e un livello alto, da scegliere secondo le caratteristiche del circuito audio di uscita. Prevede inoltre un segnale per pilotare il motore del registratore che può essere quindi messo in moto da programma. Questo particolare deve essere controllato dall'utilizzatore perché la corrente di commutazione potrebbe caricare eccessivamente l'alimentatore dell'AIM e quindi essere consigliabile adottare una soluzione che prevede il pilotaggio di un relè con il segnale dell'AIM in modo da disaccoppiare la periferica.*

*Data la necessità di disporre di maggiore RAM, oltre che di avere a disposizione altre periferiche, quasi subito sono apparse delle schede di espansione che l'AIM65 ha in qualche caso condiviso con il KIM-1, altro sistema single board basato sul 6502 e quasi compatibile, come affermato poco sopra. Le più diffuse sono state prodotte da una certa MTU come ad esempio la scheda "visible memory" o la scheda grafica per pilotare un monitor. E' stato prodotto addirittura un sintetizzatore musicale oltre ad altre cosucce come programmatori di EPROM, box di espansione,*

*vari sensori e attuatori esterni utili in esperimenti e per tenere sotto controllo parametri fisici come temperatura, tensione, etc…*

*Non è un caso che la piastra AIM65 sia stata usata spesso come supporto ad esperimenti scientifici che prevedevano la misurazione e l'elaborazione dei dati; soprattutto si vede apprezzata la possibilità di pilotare i registratori con comando di accensione e spegnimento dei rispettivi motori di trascinamento cassette. Pare niente ma immaginate di dover immagazzinare il risultato di misure periodiche fatte da qualche strumento interfacciato con il vostro AIM65; potreste, ogni tanto, accendee uno dei registratori e registrarvi i dati letti e magari nei tempi morti accendere l'altro per leggere ed elaborare altri dati...*

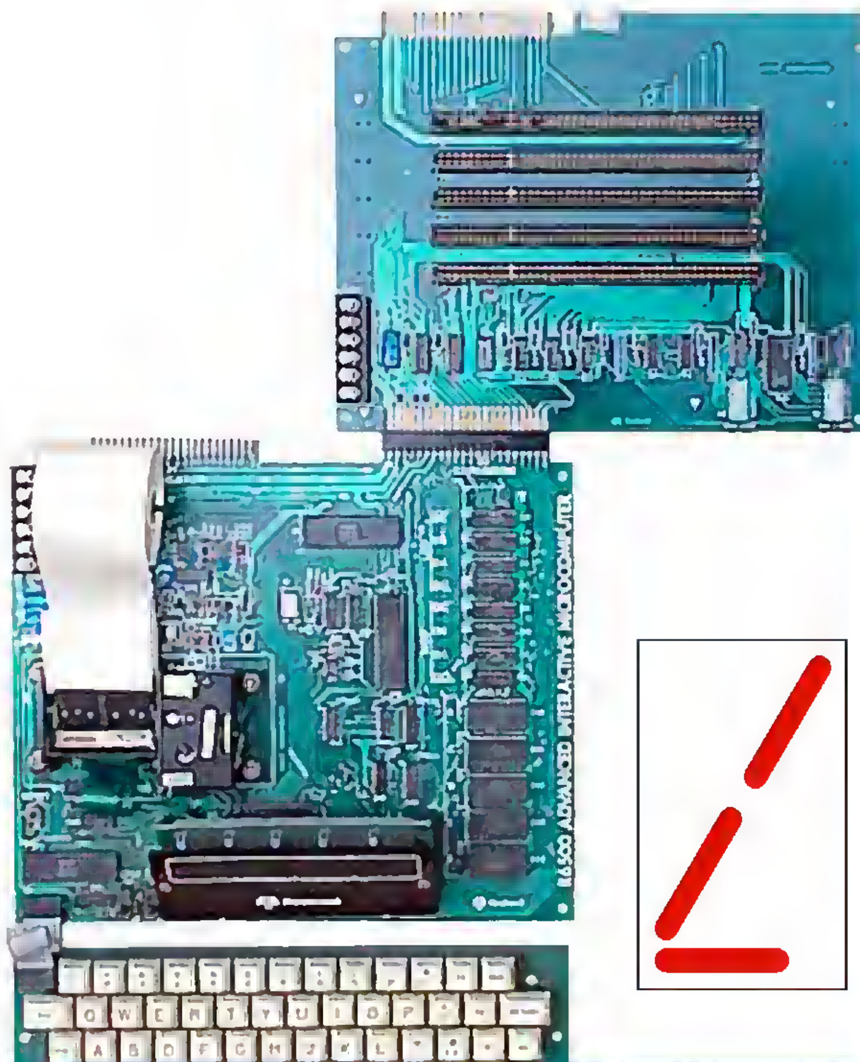
## *Espansioni*

*Ne sono state progettate e commercializzate parecchie, principalmente dalla stessa Rockwell, ma anche molte di origine più obbistica.*

*Quasi indispensabile l'estensione del bus in grado di accogliere più schede in una configurazione meccanica. Si tratta del "AIM 65 Expansion Motherboard", costruito ufficialmente dalla Rockwell è in grado di ospetare tutte le schede costruite dallo stesso produttor.*

*Una curiosità l'espansione che ospita una memoria a bolle da 64 Kbytes, una tipologia di storage non volatile che sembrava potesse avere un futuro più radioso di quello che gli è toccato nella realtà.*





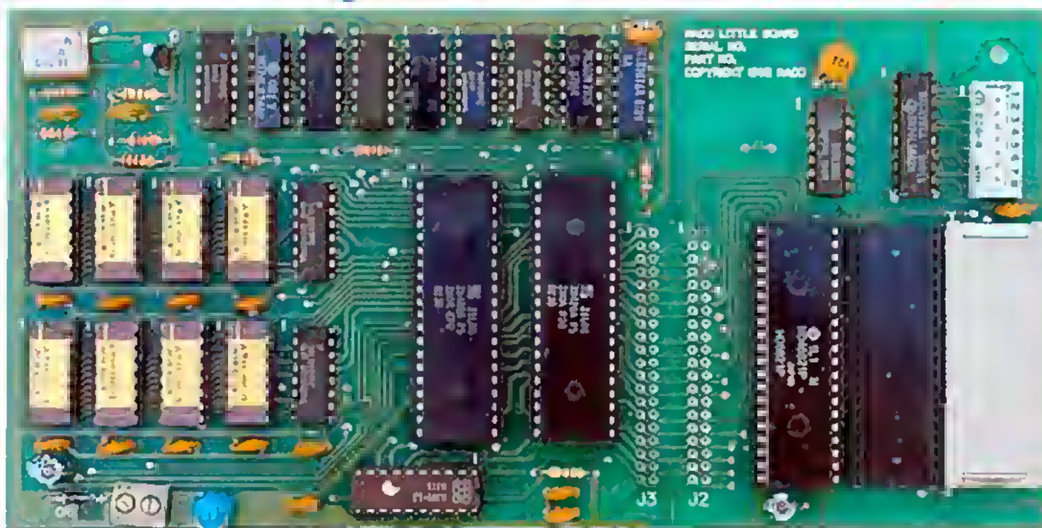
Piastra base più bus di espansione (un insieme decisamente ingombrante...).
Nell'inserto il carattere prompt "<".
(Sotto) la scheda Z80 per usare il CP/M e relativi programmi.

Addirittura si è prodotta da una certa RACO, che starebbe per Rockwell Anaheim Computer Organization, una scheda con Z80 e relativo CP/M (foto in fondo alla pagina). Evidentemente una scorciatoia per dotare il sistema di un ambiente applicativo compatibile con molto software e trasformare quindi l'AIM65 da sistema di sviluppo in sistema di elaborazione personale. La scheda ha a bordo 64 Kb di RAM e una PROM da 256 byte in grado di "boot-are" dal disco il relativo CP/M.

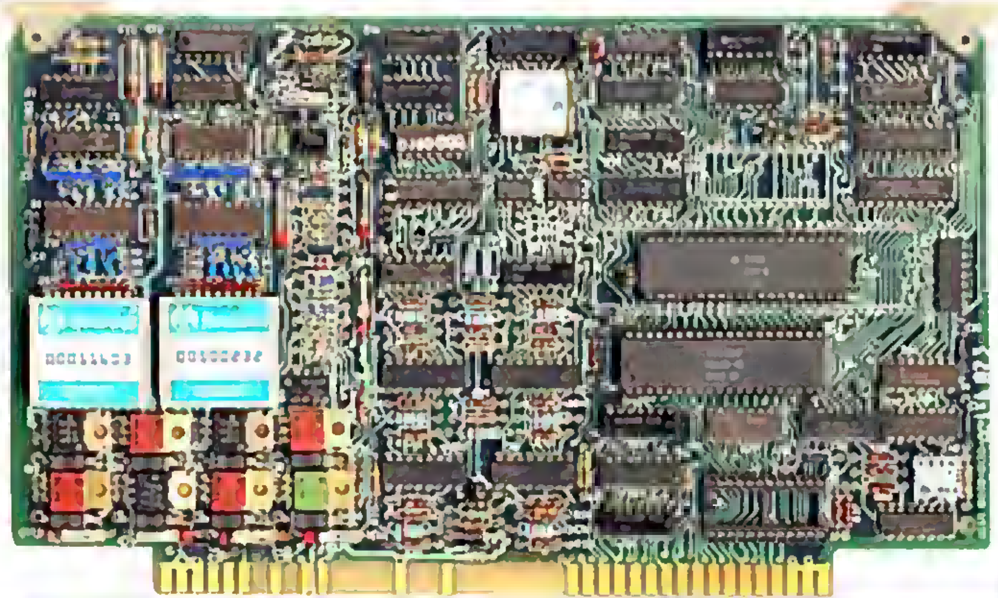
## Uso

Dopo aver connesso l'alimentatore ci si deve assicurare che lo switch KB/TTY sia in posizione Kb (tastiera) e che lo switch (RUN/STEP) sia in posizione RUN. A questo punto si può dare tensione e verificare che la procedura di boot sia eseguita correttamente.

Dopo un lampeggiamento del display se tutto è OK, il sistema stampa e visualizza sul display il messaggio "ROCKWELL AIM 65". A questo punto con il tasto <ESC> si attiva il monitor di sistema che ha come prompt il carattere '<' (non proprio questo, ma nell'intenzione…), che si trasformerà in '>' dopo l'inserimento del primo carattere in modo che il singolo carattere iniziale della riga rimanga isolato ad individuare un comando dato al monitor.



A questo punto sono attivi i comandi del ricco monitor di sistema che permettono un discreto controllo delle periferiche e dell'esplorazione della memo-



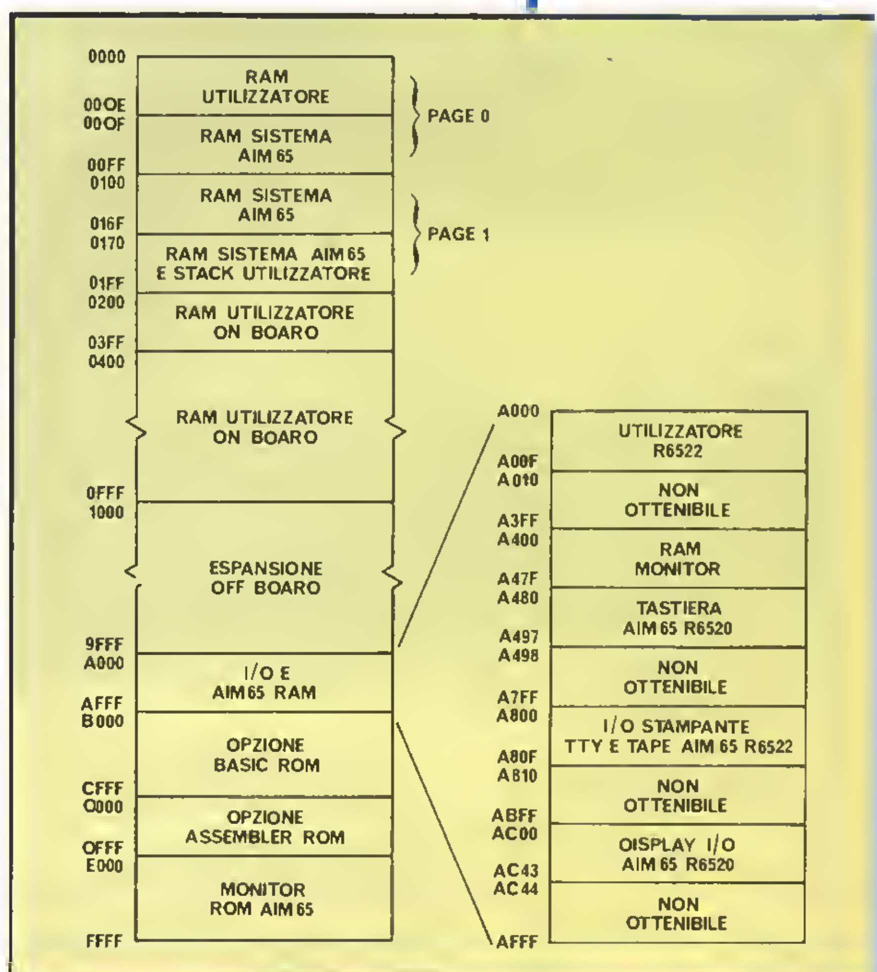
*ria ed esecuzione dei programmi residenti in ROM.*

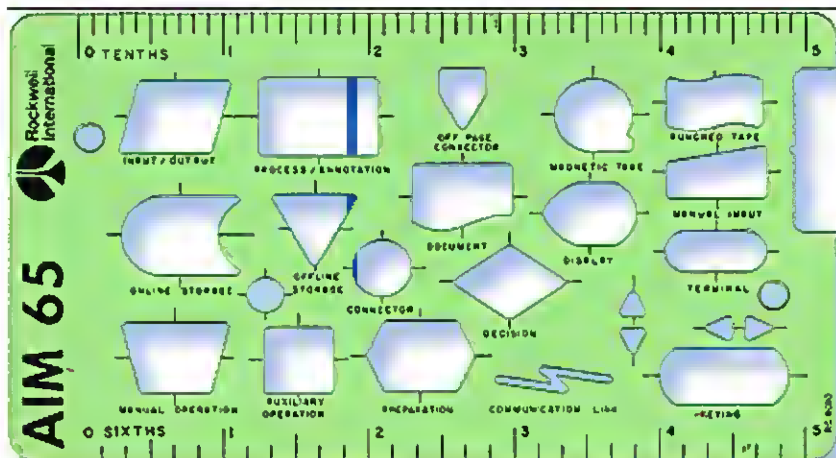
*Scopo principale di chi acquista l'AIM65 è probabilmente quello della programmazione in linguaggio macchina. Ecco dunque che l'AIM65 nella configurazione "standard" mette a disposizione un editor e un assemblatore che permette di creare il sorgente usando le istruzioni mnemoniche del processore piuttosto che i codici esadecimale direttamente inseriti in ROM. Da un certo punto di vista la programmazione esadecimale diretta è più agile per chi appronta piccole routine, ma indubbiamente è più ostica e non può godere di quelle features che sono tipiche di un linguaggio più vicino all'utente. Disporre di un sorgente permette infatti di intervenire con correzioni, cancellazioni o inserimento di istruzioni senza ricorrere ai salti mortali (o alla proliferazione delle istruzioni NOP) tipici di chi si deve "arrangiare" con il codice macchina. Pensiamo ad esempio alla comodità delle label e dei salti relativi.*

*L'assembler fornito in 4 Kb di ROM è del tipo "a due passate"; nel primo step il sorgente viene scandito per creare la cosidetta "symbols table", la tabella dei simboli, nella seconda passata si produce il codice macchina pronto per l'esecuzione.*

*Con appena 4 Kb di RAM non è che si possano fare i salti mortali, ma si sa che la programmazione in assembly è adatta a piccoli progetti e non certo a migliaia di righe di codice. Disponibili all'utente non sono nemmeno tutti e quattro i Kb di memoria, dal momento che le due*

*(Sopra) la scheda di memoria a bolle e (sotto) la mappa di memoria, informazione indispensabile per chi programma con l'assembler.*






La Rockwell ha pensato a tutto: addirittura il normografo con i simboli da flowchart.

prime pagine (la zero e la 1, cioè gli indirizzi da 0x0000 a 0x01FF) sono riservate. Il microprocessore 6502 sappiamo che non è particolarmente ricco di registri interni ma ha dei modi di indirizzamento in pagina zero (i primi 256 byte di memoria) che di fatto permettono di usarla come dei veri e propri registri interni.

Troviamo interessante la possibilità di pilotare due registratori a cassetta direttamente da programma. Questo può aiutare a mettere a punto un sistema per l'acquisizione automatica di segnali, magari a fronte del verificarsi di una qualche situazione che genera un interrupt, segnali che vengono poi scaricati su cassetta per la successiva elaborazione. Questo tipo di utilizzo della piastra della Rockwell è stata riportata in numerosi articoli apparsi su riviste specializzate attorno agli anni '80.

Da non sottovalutare nemmeno la disponibilità di linguaggi di alto livello come il principe di tutti: il BASIC. Il BASIC a corredo (opzionale) dell'AIM65 è una versione Microsoft standard, secondo la tradizione della ditta di Bill Gates, con la possibilità aggiuntiva di pilotare la stampantina interna e programmarsi delle entry per la gestione delle periferiche eventualmente usate come espansione.

Per quanto riguarda l'utilizzo di altri interpreti/compilatori riteniamo che non siano stati in testa ai desideri di acquisto da parte degli appassionati. Ogni software aggiuntivo, anche per il fatto che viene rilasciato su ROM, ha un certo costo: dalle 100 alle 150 mila lire in Italia, il che non è proprio pochissimo per una macchina che originariamente si aggira attorno alle 600.000 lire dall'importatore ufficiale.

Pieghevole con le istruzioni e i comandi dell'interprete BASIC.

ARITHMETIC FUNCTIONS

TABLE OF ERROR CODES

MESSAGES

ZERO PAGE PARAMETERS

BASIC INSTRUCTION FORMAT

Rockwell
AIM 65 8K BASIC
Reference Card

Commands

Input/Output

String Functions

Program Statements

Arithmetic Functions

Rockwell Microelectronic Devices
Sales Offices

DOC. NO. 29650N56
MAY 1979

## Software

Dopo aver trattato dell'Assembly, linguaggio assolutamente d'elezione per un sistema di

*sviluppo di questo genere, viene la naturale curiosità di esplorare le possibilità di programmazione offerte dagli altri idiomi messi a disposizione dalla Rockwell. In particolare il BASIC e il PL/65.*

*Per quanto riguarda il BASIC, si tratta di un classico "alla Microsoft"*

*Un linguaggio interessante, perché "esoterico" rispetto al classico BASIC, è il PL/65, creato dalla stessa Rockwell al fine di mettere a disposizione un agile idioma per la programmazione di interfacce di controllo cui l'AIM65 costituisce il core.*

*Si tratta di un mix fra il PL/I e l'ALGOL, due linguaggi implementati da alcuni anni sulle macchine mainframe e mini dipartimentali. Non è il caso di dilungarci troppo nella descrizione dei due linguaggi "genitori" del PL/65, peraltro ben documentati su Internet, tuttavia un piccolo assaggio è utile per capire le origini dell'implementazione fatta dalla Rockewell.*

*Ecco un esempio di procedura scritta il PL/I:*

```
FINDSTRINGS: PROCEDURE
OPTIONS(MAIN);

DECLARE
PAT VARYING CHARACTER(100),
LINEBUF VARYING
      CHARACTER(100),
(LINENO, NDFILE, IX)
      FIXED BINARY;

NDFILE = 0;
ON ENDFILE(SYSIN) NDFILE=1;
```




```
GET EDIT(PAT)  (A);
LINENO = 1;
DO WHILE  (NDFILE=0);
   GET EDIT(LINEBUF)  (A);
   IF LENGTH(LINEBUF) > 0
   THEN DO;
   IX = INDEX(LINEBUF, PAT);
   IF IX > 0 THEN DO;
      PUT SKIP EDIT
     (LINENO,LINEBUF)(F(2),A);
   END;
   END;
   LINENO = LINENO + 1;
END;
END FINDSTRINGS;
```

*La funzione FINDSTRINGS riceve in input (file sysin) una serie di righe di testo e le stampa in output precedute dal numero di riga.*

*Nel PL/I è molto interessante ed appare forse per la prima volta l'idea di trattare l'input come flusso di byte registrati su file.*

*Per quanto riguarda l'ALGOL vediamo qui un piccolo esempio:*

```
procedure Absmax(a)
Size:(n, m) Result:(y)
Subscripts:(i, k);

value n, m;
array a;
integer n, m, i, k;
real y;
```

*Oltre alla documentazione "nativa", l'R6500 ha "goduto" di numerose monografie che direttamente o indirettamente ne hanno esaminato e presentato le cvaratteristiche.*




Le riviste di informatica in Italia hanno cominciato ad uscire proprio quando l'AIM65 era sul mercato. Qui la copertina del numero 2 della rivista "m&p microcomputer" che contiene la recensione assieme a quella dell'APple ][. La data è il 1981.

```
begin
   integer p, q;
   y := 0;
   i := k := 1;
   for p:=1 step 1
until n do
    for q:=1 step 1
until m do
          if abs(a[p, q]) >
y then
          begin
           y := abs(a[p,
q]);
           i := p;
           k := q
         end
end Absmax
```

*La funzione Absmax(a) riceve in input una matrice nxm di numeri interi, cerca il massimo assoluto e lo restituisce in output nel parametro Result, assieme alle coordinate (i,j) dove l'elemento risiede.*

*Vediamo come l'ALGOL assomigli molto al Pascal, che infatti da esso deriva.*

*Il PL/65 mixa l'uno e l'altro per ottenere da un lato la strutturazione dell'ALGOL e dall'altra il paradigma di basso livello per l'I/O.*

*Esempio bubblesort scritto in PL/65:*

```
PAGE '**SORT**';
  ;
  DECLARE F, I, TMP;
  ENTRY $200;
  ;
  N=N-2;
  "SET TERMINAL VALUE FOR
 LOOP";
   F = 1; "SET FLAG";
   WHILE F = 1 DO;
     F = 0;
     FOR I = 0 TO N BEGIN;
      IF B[I] > B[I+1] THEN
      BEGIN;
       F = 1;
       TMP = B[I];
       B[I]=B[I+1];
       B[I+1]=TMP;
      END;
     END;
   END;
   N: DATA 10;
   B: DATA 23,55,36,28,54,39,9
 9,86,21,67;
   ;
   EXIT;
```

*Si vede come il linguaggio sia stato adattato per un sistema di sviluppo. Ad esempio lo statement:*

```
ENTRY $200;
```

*dichiara l'indirizzo iniziale in memoria del codice oggetto.*

*Interessante anche la dichiarazione e caricamento immediato delle variabili (N e B), in coda al codice. In questo caso N contiene il numero di elementi da ordinare e B è il vettore unidimensionale di interi da ordinare.*

*Il PL/65 è talmente orientato al processore 6502 che addirittura le variabili non possono chiamarsi A, X, Y, P e S, che sono i nomi dei registri interni al processore. Alcune istruzioni agiscono a livello di bit e richiamano direttamente le corrispondenti implementate nel micro codice del processore: SHIFT, ROTATE, CLEAR, SET, CODE.*

*Per assegnare un valore intero short (un byte) ad una variabile si possono usare le istruzioni di assegnazione diretta o indiretta ed inoltre si possono forzare le locazioni in pagina zero per la variabile stessa. La pagina zero è molto importante per il 6502, il quale può accedervi con istruzioni di due byte, più veloci rispetto all'accesso alle altre pagine di memoria.*

*La cosa funziona così: se si premette il simbolo "@" al nome di variabile, significa che vogliamo che sia immagazzinata in pagina zero. Se il carattere prefissato è invece "&", allora la variabile sarà immagazzinata nel classico stack*

*predisposto dal compilatore in una pagina di memoria "alta".*

*Ad esempio:*

*@T = 5*

*&Z = 6.*

## *Conclusioni*

*Il sistema di sviluppo R6500 della Rockwell è tutto ciò che l'appassionato hobbista e il progettista di circuiti di controllo e automazione, possano desiderare. Completo, discretamente espandibile e abbastanza economico per le caratteristiche che lo contraddistinguono, esso si pone al vertice dei sistemi di sviluppo per microprcessore che hanno avuto una impensabile diffusione prima del 1980.*

*Con la sua tastiera alfanumerica completa, il display e la stampante in grado di mostrare più dei classici e angusti display da 6/8 nexie dei sistemi concorrenti, detta decisamente un nuovo standard in laboratorio. Non è azzardato ipotizzare che la fortuna del microprocessore 6502 derivi per buona parte dalla diffusione dei sistemi di sviluppo e dell'AIM65 in particolare.*

*[Sm]*

## Bibliografia


**Wikipedia:**
http://en.wikipedia.org/wiki/AIM-65

**Wapedia:**
**http:**//wapedia.mobi/en/AIM_65
http://wapedia.mobi/en/PL/I
http://wapedia.mobi/en/ALGOL

**Recensioni**
m&p computer n. 2
Bit n. 16, aprile 1981
http://www.atarimagazines.com/compute/issue1/3229_1_AIM_65_REVIEW.php
http://www.dhub.org/object/164577
http://www.atarimagazines.com/computeii/issue1/page40.php
http://www.regnirps.com/Apple6502stuff/r65c02.htm

**Collezioni**
Old Computer Museum: http://www.oldcomputermuseum.com/aim_65.html
Power House Museum:
http://www.powerhousemuseum.com/collection/database/?irn=164576
The Freeman PC Museum Collection:
http://www.thepcmuseum.net/details.php?RECORD_KEY(museum)=id&id(museum)=784&PHPSESSID=2eec6b0ffaa7e0671f0ae923ceaa9b5c
Vintage computers:
http://www.vintage-computer.com/aim65.shtml
Old computer net:
http://oldcomputers.net/AIM-65.html
Old-Computers:
http://www.old-computers.com/museum/computer.asp?st=1&c=58

**Manuali:**
AIM65 programming chart
PL/65 user's manual
AIM 65 BASIC language reference Manual
AIM 65 Schematic poster
AIM 65 FORTH manual
AIM 65 User's Manual
AIM 65 Monitor Guide
MCS6500 Istruction set Summary
R6500 Hardware Manual

**Riviste dedicate:**
Interactive issue 01-08; Rockwell International

# Come eravamo...

La storia dei sistemi e degli uomini che hanno creato un mondo nuovo.

## Storia dell'interfaccia utente (3)

### Gli albori dell'interfaccia grafica

Nelle due puntate precedenti di questa breve storia dell'interfaccia utente, abbiamo visto una rassegna di metodi e apparecchi progettati allo scopo di consentire una comunicazione efficace fra uomo e macchina. E' venuto ora il momento di addentrarci in quella che oggi viene considerata la "vera" interfaccia utente, cioè quella che fa uso della grafica per impostare il paradigma della scrivania.

Fin dall'inizio degli anni settanta, le aziende più attive nel settore dell'elaborazione dati, stavano studiando la possibilità di dotare i sistemi di una interfaccia grafica a pixel. Le motivazioni si ritrovano soprattutto per il fatto che erano nate le Workstation, un sistema molto simile nelle prestazioni ad un mini dipartimentale, ma dedicato all'uso personale. Avere a disposizione una macchina di classe workstation era costoso e se ne giustificava l'adozione solo a fronte di elaborazioni molto particolari e "pesanti", come la grafica industriale, appunto.

Capofila di questi esperimenti è stata la Xerox Corporation, ma non la sola, anche se è rimasta questa ditta nell'immaginario collettivo, grazie sostanzialmente alla leggenda secondo la quale Steve Jobs ne avrebbe rubato i segreti per progettare l'interfaccia del Lisa prima e del Mac subito dopo.

Il primo tentativo di Xerox si concretizza nel prodotto denominato Alto (vedi figura 1). Non è proprio quello che i tecnici volevano ma ci stavano lavorando...

Tralasciando i tentativi più o meno riusciti che sono seguiti a questo primo esperimento, vale comunque la pena citare la Three Rivers Computer Corporation, che nel 1980 mette in commercio una workstation grafica chiamata Perq Graphical Workstation (figura 2)

Saltiamo subito al 1981 quando la Xerox rilascia il suo sistema gra-



Figura 1.
La workstation "ALTO" del 1973 ad opera dei laboratori Rank Xerox di Palo Alto in California. Siamo ovviamente nella mitica Silicon Valley.

fico Star (Figura 3).

Si tratta di una workstation con display a toni di grigio con icone, finestre sovrapponibili e ovviamente mouse.

Incomincia a delinearsi la strada maestra da seguire: grafica, puntatore con mouse e finestre come contenitori dei task in esecuzione sul sistema.

Il sistema Star di Xerox si dice comunemente sia il papà (putativo) del primo Mac, cioè il Lisa di Apple, uscito nel 1983 e che ha sparigliato le carte rispetto al mercato dei sistemi personali (Figura 4).

Il Lisa si andava a mettere nella fascia di prezzo a metà strada fra il costo del personal e la workstation scientifica più sofisticata. Purtroppo sappiamo che l'idea era buona ma il momento non favorevole. Il secondo e più abbordabile tentativo di Apple avvenne con il Mac Hintosh nel 1984, solo pochi anni dopo il Lisa ma una "vita" distante in termini informatici.

Il sistema operativo Mac conserva gli elementi fondamentali di Lisa ma si orienta verso l'utilizzo per un utente meno esperto e in possesso di una piattaforma elaborativa migliore.

Lisa, aldilà della novità tecnologica e dell'interazione grafica, ha introdotto un concetto fondamentale che poi Apple non abbandonò mai: il paradigma dell'iterazione ad oggetti. Ogni elemento a video risponde alle stesse azioni (ad esempio al click del mouse) in maniera differente, è cioè polimorfico. Per l'utente è un grande vantaggio perché non si deve chiedere cosa ci stia sotto l'oggetto ma interagire con esso esattamente come farebbe con un oggetto fisico deposto sulla sua scrivania.



Figura 2.
La worstation grafica della Three Rivers.

Il PC IBM, rilasciato nel 1981, ha avuto vari tentativi nel senso grafico.

Uno di questi è VisiOn della Visi Corporation (quella che produceva il Visicalc). VisiOn ha lo stile mac-iano ma cerca di differenziarsi almeno per il posizionamento del menù, che si trova sul piede di ogni singola finestra; quando invece il menù generale stile Mac, che

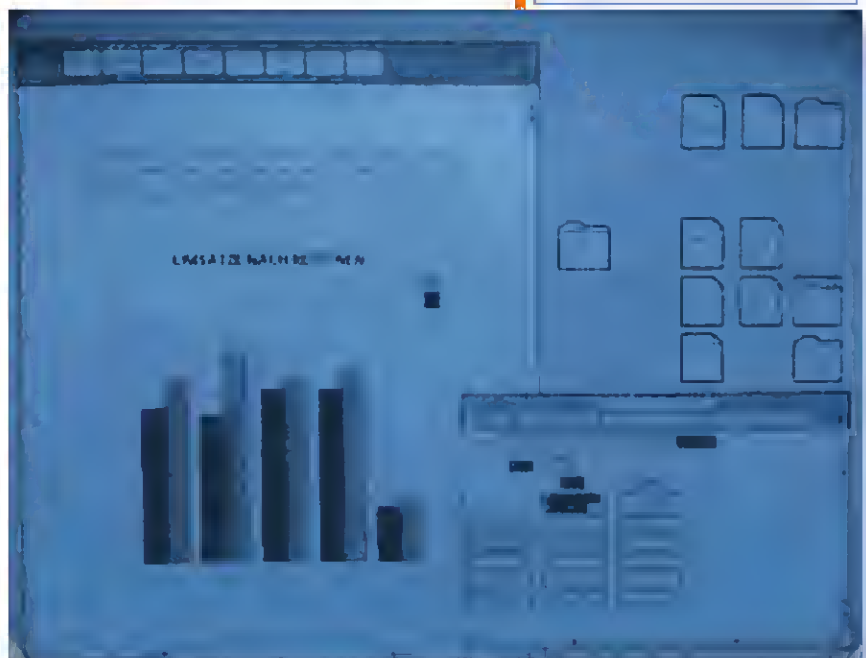


Figura 3.
La grafica del sistema Star di Xerox.




*Figura 4.*
*La scrivania virtuale del Lisa di Apple.*

*si trova nella parte alta del video, è quanto più comodo e naturale si possa concepire, ma sembra l'abbia capito solo Apple (Figura 5).*

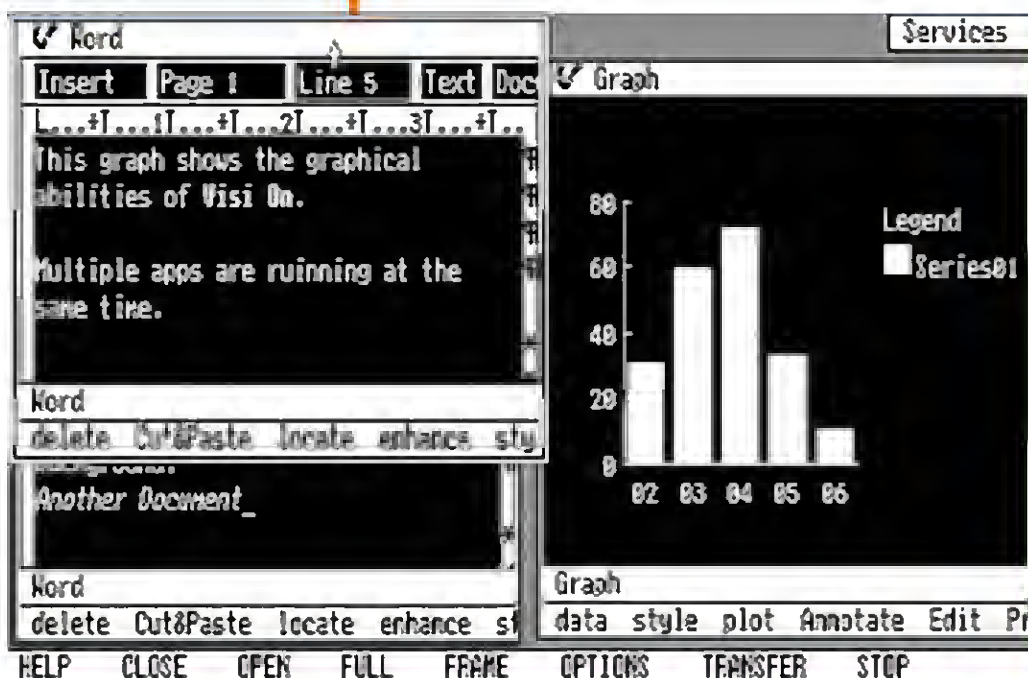



*Figura 5.*
*VisiOn, la proposta di Visicorp. SI noti l'idea primitiva del menù nel piede della finestra, una pratica che non ha preso piede (è il caso di dirlo!)*

*E siamo giunti così al 1984, anno Orwelliano dalle inquietanti predizioni... Il 1984 è un po' l'anno d'oro delle interfacce grafiche. Il Mac di Apple trova la sua veste definitiva con un maggiore sobrietà che si trasforma in eleganza (figura 6).*

*Dopo il Mac e la sua ecclatante presentazione pubblicitaria costata una piccola fortuna all'azienda di Jobs e compagni, esce GEM della Digital Research (quella del CP/M), che crea un ambiente portabile (ne esistono versioni per Commodore 64, Atari ST, etc…), basato su icone e finestre sovrapponibili.*

*GEM è quasi un Mac-clone e nemmeno tanto camuffato…(figura 7).*

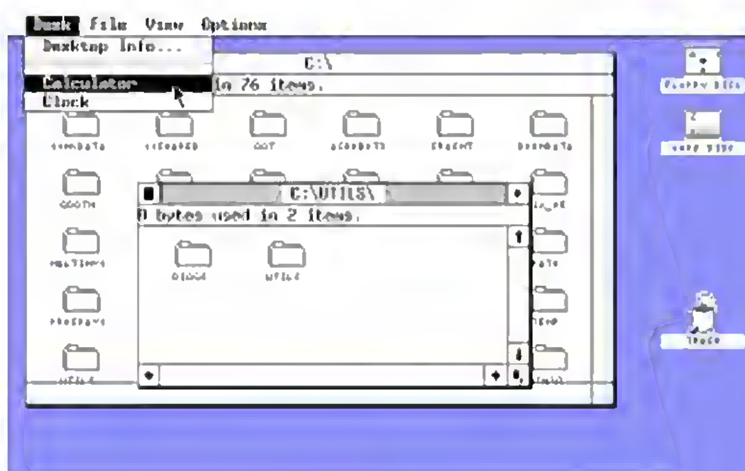



*Figura 7.*
*Il GEM ha conosciuto una discreta fortuna, come si vede meritata.*

*Un'altra pietra miliare, siamo sempre nel 1984, è l'uscita del sistema X Window del MIT.*

*Confezionato in diverse "taglie" il sistema X Window si adatta ad ogni tipo di terminale e può essere configurato da un utente con qualche capacità tecnica. Gira sotto Unix e successivamente diventerà l'ambiente grafico di GNU/Linux.*

*X Window è tutt'altro che un ambiente monolitico. Già abbiamo accennato alla possibilità di configurarlo attraverso dei file di testo, ma anche di appiccicarci sopra una libreria grafica che vada ad occuparsi dell'aspetto delle finestre e degli altri elementi. Una di queste librerie, famosa per diffusione, è stata Moo-tif, adottata da Sun Corporation per la linea delle suo Workstation grafiche basate sullo Unix BSD.*

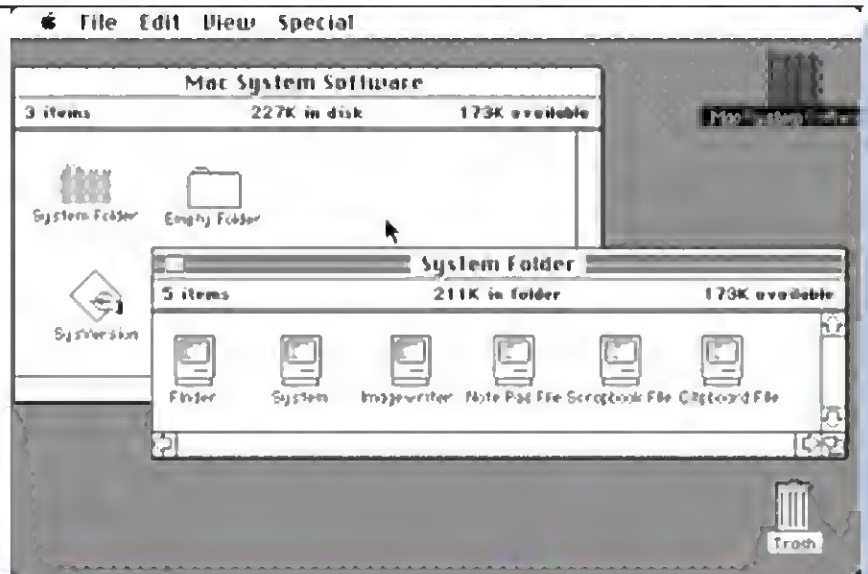



*Figura 6.*
*La scrivania Mac nella sua versione 1984 ma che rimarrà praticamente la stessa per lungo tempo (a parte l'introduzione del colore).*

*Il 1984 volge al termine ma è tutto un fermento. Cosa ci riserverà il futuro? Tutti parlano di qualcosa che sta preparando la Microsoft... Staremo a vedere.*

***[Tn]***

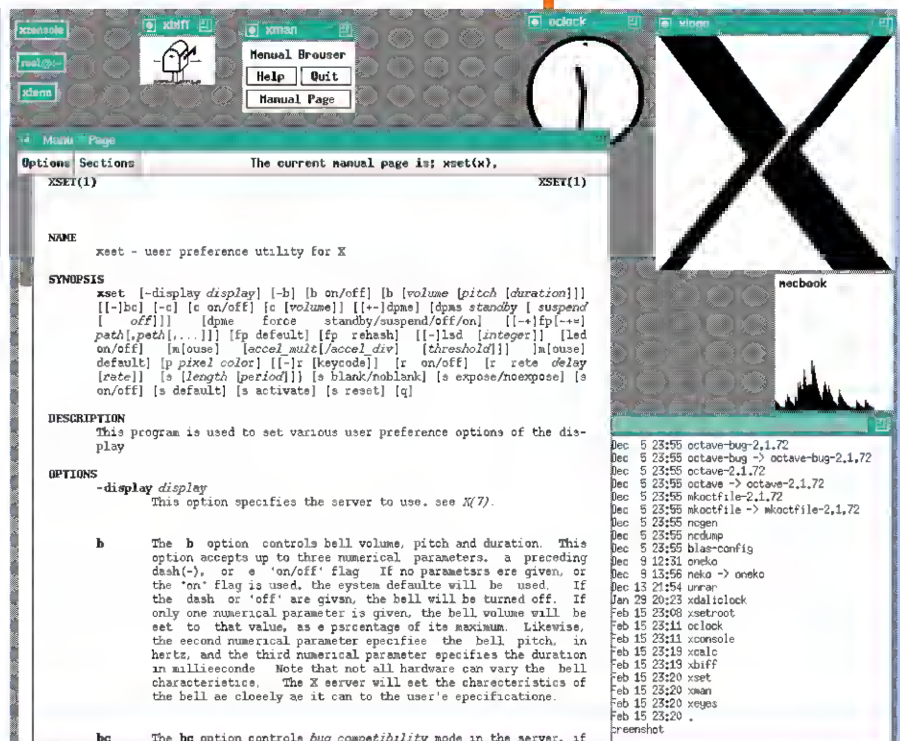



*Figura 8.*
*Xwindow, ovvero la soluzione Unix al problema della interfaccia grafica.*

# Il racconto

Rubrica curata da [Bs]

Storie di vita dove i computer (soprattutto retro computer) c'entrano in qualche modo.

## Automatik(4) - Il nuovo lavoro

*Dove si racconta come divenni finalmente "tecnico elettronico".*

*Era uno degli ultimi giorni della mia esperienza di stage e il mio futuro datore di lavoro era venuto per tirare le fila e decidere se ero adatto o meno alla sua azienda. Credo che nella decisione di prendermi e nel parere favorevole del sig. Tiziano, pesasse più quella giornata di facchinaggio dei biliardi piuttosto che la fiducia nelle mie capacità tecniche, ma tant'è fui promosso con l'invito a presentarmi di li a quindici giorni nell'azienda che sarebbe diventato il mio posto di lavoro nei prossimi due anni.*

*Era il mio turno di decidere: ci dovevo andare lasciando il mio lavoro in fondo comodo di commesso sulla soglia di casa per fare ogni giorno venti più venti chilometri di macchina e trovare anche il modo di mangiare a mezzogiorno. Se avessi avuto tutte le carte in mano per decidere probabilmente sarei rimasto dov'ero: ci stavo bene, guadagnavo poco ma in manera sicura, avevo ferie e permessi pagati e un sacco di tempo libero proprio per la tipologia di orario.*

*Sì, mi pesava lavorare di sabato e di domenica, ma le mezze giornate di chiusura, ricordo erano il lunedì e il mercoledì, mi permettevano di seguire molti interessi. Avevo naturalmente tentato di proseguire con l'Università ma presto mi resi conto che non era possibile. Ammiro incondizionatamente chi ci riesce; io non ero fra questi.*

*Ma ci stavo stretto, l'ho già detto, in quel supermercato di paese e sapevo che sarei potuto crescere sono seguendo una carriera che non mi attirava, sostanzialmente quella del direttore di un reparto prima e di un negozio poi. Poi c'era l'informatica, materia del tutto nuova nei primi anni '80 e che mi aveva appassionato e convinto che li c'era da lavorare e che non dovevo perdere il treno dell'innovazione se volevo far fruttare un poco le mie doti di intelligenza.*

*Decisi quindi di licenziarmi e firmare il contratto per la nuova avventura dove io mi qualificavo "tecnico elettronico", ma sul libretto di lavoro, nella colonna "qualifica" figurava un più umile "operaio".*

*Il mio nuovo datore di lavoro si chiamava Romano ed era una persona buona di animo, ne sono convinto, però anche lui era soggetto alla malattia che fa incrinare l'onestà dei piccoli imprenditori-commer-*

*cianti-artigiani: l'idea che siccome loro ti pagano, cioè tu mangi grazie a loro, li autorizza a trattarti come una cosa loro, senza molti diritti.*

*Nella mia ingenuità avevo discusso solo l'importo mensile del salario, decidendo con Romano che sarebbe stato pari al posto che lasciavo, e dell'orario di lavoro alla cui mia domanda Romano aveva risposto semplicemente con "la mattina si comincia alle nove e il sabato normalmente non si lavora".*

*Il lavoro che lasciavo era normato in maniera completa: orario, ferie, malattie, scatti, pensione, straordinari,... insomma tutto era chiaro e trasparente e non capii immediatamente che le cose sul nuovo posto di lavoro potevano essere diverse, molto diverse. Quando cominciai a realizzare che mi ero messo in una trappola era troppo tardi: avrei potuto tornare indietro, rinunciare al lavoro e ripresentarmi sul mio vecchio posto di lavoro implorando di essere ripreso? Sì, l'avrei potuto e con la mia esperienza di oggi lo farei, ingoiando il mio orgoglio assieme al mio amor proprio. Ma allora ero giovane, fiducioso, credulone... Una combinazione di dis-qualità foriere di guai e nemiche della vita stessa.*

*Romano non fu del tutto onesto nei miei confronti, un po' mi dispiace affermarlo, ma non posso dire che lo fu. Prima di tutto il contratto non lo firmai il primo giorno ma una settimana dopo (il commercialista non l'aveva ancora preparato), poi scoprì che l'assunzione era datata il primo giorno del mese successivo ("per ragioni di ordine, ma non ci avrei perso nulla"), infine non poteva non dirmelo che sarei stato assunto formalmente per quattro ore al giorno ("le tasse sono così care!").*

*Rimasi negativamente impressionato dalla notizia e dalla manfrina che tirò giò Romano sul fisco che strozzava la libera iniziativa e che lui non poteva permettersi di "mettermi in regola" per otto ore, etc...*

*Che potevo fare? Buttai giù quel boccone amaro cercando di non pensare che stavo mettendo a repentaglio il mio futuro. Quando si è così lontani dalla pensione non ci si pensa molto che un mese di contributi persi da giovani si traduce in un mese di fatica da vecchi...*

*L'orario di lavoro è vero che iniziava alle nove, ma quando finiva lo sapeva solo il cielo! Normalmente alle sei del pomeriggio si poteva andarsene, ma questo era vero solo se il titolare non era presente, altrimenti trovava sempre qualcosa da fare, almeno fino alle sette, ora in cui andava a cena. Senza contare che spesso eravamo in giro per installare giochi nei bar e ritirare gli incassi e ovviamente ci si rimaneva fino a che tutto era a posto e il giro completato.*

*La media giornaliera era di nove ore, ma non si timbrava e quindi tutto andava "in cavalleria". E' vero che Romano ti riconosceva un plus in nero sullo stipendio, ma evidentemente conveniva a lui più che a noi. Il sabato "di norma" era libero,*

*questo aveva detto il titolare nei colloqui prima dell'assunzione. Mi sembrava logico che una azienda che viveva di incassi di macchinette da gioco potesse chiedere un impegno straordinario ai suoi dipendenti piuttosto che lasciare guasta una macchina durante il week end. Il problema era che queste emergenze erano costanti e almeno la mattina si lavorava sempre, nel pomeriggio invece qualche volta si staccava. Per provare e per ripicca inventai delle scuse rispetto alla mia necessità di essere libero di Sabato. La prima volta me lo concesse, la seconda fece una smorfia e la terza mi disse chiaramente che lì, in azienda, tutti lavoravano di Sabato e che lui non poteva permettersi, etc, etc....*

*Durante il week end e le assenze in genere il titolare non si faceva scrupoli di chiamarti a casa e chiederti, sempre molto gentilmente, di "andare a dare un'occhiata" in quel certo Bar perché il gioco era fermo o le palline del calcetto non uscivano (normalmente i ragazzi ci mettevano dentro palle di carta per bloccare le porte e non far entrare la pallina o ci giocavano addirittura con le palline fatte di carta e elastico). Succedeva anche che il il titolare del locale chiamasse perché aveva bisogno di moneta ed era una molla che faceva scattare Romano: neiente moneta uguale niente soldi nelle macchine uguale niente guadagno. Una tautologia, direi.*

*Dopo un po' di tempo i gestori della zona mi chiamavano direttamente a casa, sia la domenica che di sera e quanto valesse questa presenza disponibile ne ebbi una misura poco dopo quando lo stesso titolare mi disse con soddisfazione che si vedeva che ci tenevo al lavoro perché gli incassi erano superiori a quelli che si riusciva a fare prima negli stessi esercizi. Ne fui compiaciuto, non lo nego: per tradizione famigliare il lavoro è sempre stato sacro in casa nostra. Ormai ero nel giro e mi conveniva ballare facendo buon viso alle magagne che il mio rapporto di lavoro si portava dietro e ricavare da esso i vantaggi che mi convenivano.*

*Mi preoccupai seriamente quando con un certo giro largo di parole Romano mi disse che alla fine dell'anno mi avrebbe licenziato per riassumermi all'inizio di gennaio. Diceva che mi conveniva: avrei ritirato il TFR subito e non so quale fosse la sua di convenienza, ma naturalmente una doveva pur esserci. Ricevetti quindi lettera di licenziamento verso il 10 di dicembre mentre i soldi del tfr mi sarebbero stati pagati in gennaio.*

*Confesso che dubitai che mi riprendesse, ma invece fu proprio così. Mi presentai regolarmente al lavoro in gennaio e dopo una decina di giorni mi chiamò in ufficio per farmi firmare il nuovo contratto che, guarda caso, avrebbe avuto corso dal primo di febbraio. Sembra che una qualche norma impedisse ad un datore di lavoro di riassumere una persona senza uno stacco temporale. Un paio d'anni più tardi ne parlai casualmente con un sindacalista che mi confermò nei miei sospetti: non era affatto vero; ancora una volta Romano si "tirava su le maniche", diciamo così, guadagnandoci comunque sempre.*

*Quindi lavorai il gennaio in nero (altro mese da fare in più per la pensione), ricevendo il normale stipendio di prima ma dei soldi che lui risparmiava non pagandomi gli oneri fiscali e previdenziali non se ne fece cenno. Mi chiedo oggi cosa sarebbe successo se mi fossi fatto male durante quel mese. D'accordo i rischi erano limitati ma comunque si era spesso in giro in macchina e le scosse da filo di alimentazione scoperto erano all'ordine del giorno...*

*Mi vergognavo a recarmi dai sindacati a far controllare le mie buste paga e il calcolo del tfr che era esiguo, visto che avevo lavorato solo tre mesi ufficialmente in quel primo anno. Non mi passò nemmeno nell'anticamera del cervello che forse non avevo fatto nemmeno un giorno di ferie e che avrebbero dovuto essermi pagate. Protestare? Ora lo farei e non avrei paura del-*

*le conseguenze, ma allora avevo l'orgoglio da zittire e dimostrare ai miei genitori, che me ne avevano sconsigliato, che invece avevo fatto la scelta giusta.*

*Però il lavoro mi piaceva abbastanza; c'era da lavorare sodo la maggior parte del tempo, ma c'erano pure giornate in cui si faceva poco o nulla e si passavano interi pomeriggi a giochicchiare con qualche titolo appena riparato in laboratorio o presso i clienti.*

*Andare a zonzo in due per tutta la regione era piacevole e andavo molto d'accordo con Daniele che era un ragazzo delizioso al punto che non ricordo il minimo screzio fra noi nei due anni che sono rimasto in azienda. Quasi ogni giorno era una avventura e devo dire di aver girato tutta la provincia come mai avevo avuto occasione di fare prima. Ovviamente si bigiava quel poco che ci era consentito, rientrando in ditta dopo le cinque anche se alle quattro avremmo potuto esserci tranquillamente. In qualche modo bisognava pur difendersi: fosse stato per Romano avremmo fatto dodici ore di lavoro al giorno! E Daniele ne conosceva di trucchi! Che ingenuo ero al suo posto: mai ero stato nella necessità di "fregare" il capo, di imboscarmi o fingere di lavorare. Ero del tutto nuovo nel "settore".*

*Ci si fermava poi a mangiare in certi ristorantini che Daniele conosceva e frequentava regolarmente durante i giri di riparazione, oppure in certi Bar dove la cameriera era così carina o nella saletta di sotto si poteva giocare a bigliardo lontani da occhi indiscreti. Parcheggiavamo allora un po' lontano, dietro la chiesa, e si passava buona parte del pomeriggio in attesa che arrivassero le cinque per rientrare.*

*I trucchi per sopravvivere erano infiniti e Daniele me li fece sperimentare tutti. All'inizio avevo una paura folle di arrossire davanti a Romano che chiedeva spiegazioni del ritardo o di altre cose. Poi ci feci l'abitudine, molto in fretta, lo confesso. Una delle tecniche preferite era fingere di non essere stati in grado di aggiustare un gioco perché ci mancava quel pezzo e si doveva ritornare nel pomeriggio. Invece il gioco era stato aggiustato ma non eravamo tanto ingenui da dirlo al titolare del Bar, casomai Romano avesse telefonato per una conferma.*

*Arrivammo addirittura a mettere delle richieste di intervento fittizie sulla segreteria telefonica con la complicità di un amico, per indurre il titolare a mandarci di urgenza il lunedì mattina in quella certa località turistica in riva al lago o in un rifugio di montagna addirittura. Un po' di rischio c'era perché Romano era per sua natura diffidente e pensandoci non mi sovviene come mai non abbia mai scoperto i nostri trucchi...*

*Girare da solo, senza il mio comagno di avventure, mi piaceva molto meno ma mi organizzai presto e con una certa abilità riuscivo a ritagliarmi qualche mezz'oretta di relax lontano dalle grinfie del capo. Portavo allora la macchina in certi posti di periferia dove trovavo spesso, imboscati quanto me, i militari delle caserme cittadine. Questi, invece che andare su e giù per fare scuola guida agli allievi, preferivano nascondere camion e camionette dietro la siepe di certi posti strategici e passare lì il pomeriggio giocando a bigliardino o a flipper e a bere qualche birra. Era facile fare amicizia con loro: per una strana combinazione avevo fatto la naia nella stessa caserma e una generosa manciata di gettoni per il flipper apriva i loro cuori alla fiducia.*

*Venne la primavera e poi l'estate e poi di nuovo l'autunno: avevo completato il giro di calendario di quel mio primo anno alla Automatik. Non ne feci di bilanci, non ero in grado e non sarei stato sereno. Mi pesava essere in regola per metà giornata e dovevo farlo presente. Ma un'altra idea stava maturando e fu quella che mi guidò su una strada diritta.*

*Ma questa è un'altra storia.*

***[Lp]***

# Edicola

## VCF Gazette

*In edicola o sul Web le riviste che parlano di computer, preferibilmente retro o free*



ENGLISH DEUTSCH ESPAÑOL FRANÇAIS ITALIANO NEDERLANDS PORTUGUES EVENTS HOME CONTACT

Vintage Computer Festival

**VCF Gazette**
**A Newsletter for the Vintage Computer Festival**

Events
Blog
Marketplace
Library
Gazette
Gallery
Projects
Donate
Sponsors
Press
Mailing List
Links
FAQ
Contact
Login

March 13, 2002
Volume 1, Issue 1

Hello Vintage Computer Fans! Welcome to the first issue of what is guaranteed to be an irregularly published newsletter to keep you up to date on the latest events and happenings of the Vintage Computer Festival.

There is a lot going on this year. We have two major events, VCF Europa 3.0 and VCF 5.0, and an Open House at our Oakland, California, facility.

VCF Europa 3.0

The third annual Vintage Computer Festival Europa is being held April 27th and 28th in Munich, Germany. If you have a chance to be in Munich during the last weekend of April, you should certainly attend. You'll get a taste of the vibrant computer industry and the varied computers that were produced in Europe from over ten years ago and beyond.

For more information on VCF Europa 3.0, visit:

http://www.vintage.org/2002/europa/

–or–

http://www.vcfe.de/



ENGLISH DEUTSCH ESPAÑOL FRANÇAIS ITALIANO NEDERLANDS PORTUGUES EVENTS HOME CONTACT

Vintage Computer Festival

**VCF Gazette**
**A Newsletter for the Vintage Computer Festival**

Events
Blog
Marketplace
Library
Gazette
Gallery
Projects
Donate
Sponsors
Press
Mailing List
Links
FAQ
Contact
Login

May 12, 2005
Volume 3, Issue 1

Wow! The VCF Gazette marks a milestone as it rolls in to its 3rd year. And boy are we late. Later than usual. Very late. In fact, I'm positive we're later than we've ever been. But my we're busy here at VCF central. Busier than usual. Very busy. In fact, I'm positive we're busier than we've ever been. So there you have it. Anyway, on with the news!

In This Issue:

VCF 7.0 Wrap Up
VCF 7.0 Exhibit Awards and Photo Gallery
VCF Midwest "Lite"
VCF Inaugurates Long-Term Data Archiving Standard: FutureKeep
New VCF Website Features

VCF 7.0 Wrap Up

The 7th annual Vintage Computer Festival was held on November 6-7 at the Computer History Museum in Mountain View, California. It's pretty much cliche at this point to say it was the best event yet, but it really was! Just ask the 450+ people who attended.

The main feature of the VCF this last time around was the Maze War Retrospective hosted by Bruce Damer of the DigiBarn. The authors of Maze War, which is the original "first person shooter" videogame, discussed the development of the game in the early 1970s. The Maze War server was ported to the PC by Ken Harrenstien and a Maze War network of three PCs plus an Imlac PDS-1D (heroically provided by Tom Uban) was installed at the VCF, allowing attendees to

### Scheda

*Titolo:*
*Vintage Computer Festival Gazette*

*Sottotitolo:*
*A Newsletter for the Vintage Computer Festival*

*Editore:*
*VCF Staff*

*Web:*
*http://www.vintage.org*

*Lingua: Inglese*
*Prezzo: Free*
*Primo numero:*
*Marzo 2002*

*Questa fanzine è stata pubblicata dal 2002 al 2005 con periodicità trimestrale fino all'ultimo numero che è del marzo 2005. Si tratta in totale di nove numeri pubblicati sul Web che trattano i temi del retro computer legati alla manifestazione Vintage Computer Festival. Questa organizzazione culturale mantiene un sito Web con lo scopo primario di sostenere le iniziative du ricerca e culturali relative la retro computing.*

*Primaria attività del gruppo, al quale si può aderire anche virtualmente, è l'organizzazione delle Exibitions che si svolgono regolarmente due/tre volte all'anno in varie sedi europee.*

*Peccato che non si sia ritenuto di continuare nell'iniziativa della gazzetta, non fosse altro che per*

*fissare in una pubblicazione periodica i progetti salienti dell'organizzazione. Il Web è un sostituto fantastico, senza ombra di dubbio, però sappiamo tutti quanto sia fragile in un certo senso: non paghi il dominio e sei fuori, invisibile al mondo e senza nessun passato!*

*Dobbiamo leggere questo periodico con la giusta ottica che è appunto quella della celebrazione delle attività del gruppo in una esposizione molto spartana, senza grafica o altro supporto multimediale. Per trovare la documentazione, soprattutto fotografica, è obbligatorio visitare il sito che è in linea con la grafica della fanzine ma ricco e ben organizzato.*

*VCF è un po' quello che molti appassionati spererebbero di poter realizzare anche in Italia, pur con le difficoltà obiettive dovute alla numerosità non certo stratosferica dei praticanti nostrani.*

**[Sn]**

**Vintage Computer Festival East 6.0**

September 12-13, 2009

InfoAge Science Center, 2201 Marconi Rd., Wall, N.J., 07719

Saturday: 10 a.m. – 6 p.m. *** Sunday: 10 a.m. – 5 p.m.

Tickets: $10 (1 day), $15 (both days), FREE for 17 and younger

**Special events @ VCF:**

- Keynote: RCA Computers in the 1950s
- 8-bit live music and visual concert
- PockeTerm Construction Workshop
- BASIC Programming Challenge
- Afternoons: Exhibit hall with live demos
- Technology by-the-pound book sale
- Museum guided tours all weekend long

Details frequently updated at www.vintage.org

Contact: Evan Koblentz, evan@snarc.net / (646) 546-9999

**Sponsors:**




VINTAGETECH

*La locandina dell'ultima esibizione (al momento in cui scriviamo) avvenuta nel settembre 2009.*

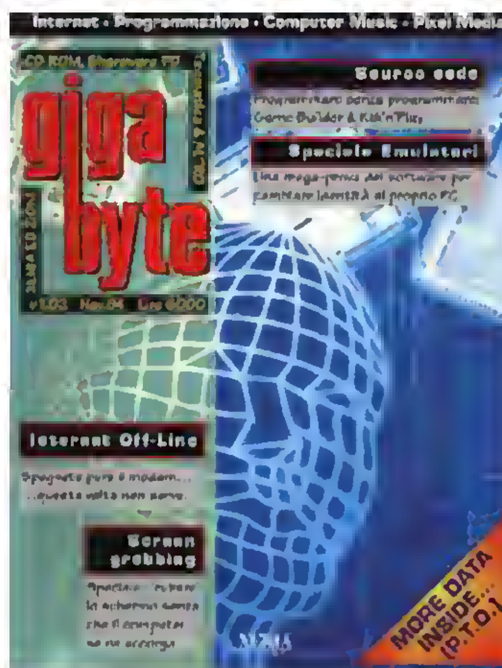
*Un angolo dell'esibizione VCF set. 2006 Si noti la cura e l'eleganza (a parte gli scatoloni sotto il tavolo) con la quale vengono realizzati i punti informativi.*



# Retro Riviste

*La rassegna dell'editoria specializzata dai primi anni '80 ad oggi*

## GigaByte




**Scheda**

*Titolo:* *GigaByte*

*Sottotitolo:* *Internet - Programmazione - Computer Music - Pixel Media*

Editore: Xenia Edizioni

*Lingua:* *Italiano*

*Primo numero:* *1993*

*Ultimo numero:* *1995*

*Prezzo* *L. 6.000*

*Edita a cura della Xenia Edizioni, la rivista uscì approssimativamente dal 1993 al 1995. Una classica meteora, insomma, come tante altre testate non solo elettroniche ed informatiche. Molto curata dal punto di vista grafico si proponeva ad un target squisitamente hobbistico concentrandosi sulla recensione di programmi per sistemi PC.*

*Devo dire di averla seguita con molto interesse anche per le qualità che ne permettevano una lettura rilassata, tipica rivista "da viaggio", ma contemporaneamente utile per ampiare la propria cultura informatica.*

*È necessario ritornare agli anni 94/95 e alla situazione dell'informatica personale di allora per comprendere l'interesse che questa rivista ha suscitato e per capire la sua prematura scomparsa. Internet è un miraggio mentre la connettività via BBS è ampliamente diffusa. Contemporaneamente esplode il mondo dello shareware: programmi da provare gratis per poi non comprarli mai (soprattutto in Italia, ma non solo). I programmi Shareware si trovano o sulle BBS o sui cd-rom che arrivano dagli Stati Uniti. Chi no ricorda ad esempio il "SimTel", una raccolta sterminata di programmi per ms-dos su due cd-rom? Qualsiasi cosa tu avessi bisogno lì la trovavi sicuramente.*

*Internet invece latita fuori delle Università. La Tecnimedia di Roma, attraverso McLink offre il collegamento ma l'utilizzo è a dir poco ridicolo. Infatti il tutto funziona via X25 (Itapac) con rifatturazione basata sul traffico a velocità 2400 bit/sec al massimo (e era già qualche cosa, visto che il modem più diffuso andava a 300 bit/sec). Insomma non è che ti venisse una voglia pazza di collegarti!*

*GigaByte si propone come recensore di programmi di ogni tipo ma principalmente dei prodotti shareware e freeware che sono così facilmente reperibili. Credo che uno dei motivi che contribuii alla diffusione del periodico sia stata la notoria scarsa padronanza della lingua inglese da parte di noi italiani e la conseguente difficoltà nel leggere i manuali allegati al software. Negli articoli trovavi tutto quello che serviva per installare e cominciare a lavorarci.*

*La decisione dell'editore di sospendere le pubblicazioni e di far confluire la redazione in un'altra rivista, sempre edita da Xenia: "PC Action", mi lasciò scontento! Non perché PC Action fosse una rivista da buttare, ma perché la trovavo "fuori sintonia" sia per il fatto che era incentrata sui giochi ma soprattutto per l'accozzaglia di argomenti impaginati senza logica e perché erano usati degli sfondi colorati per le pagine che rendevano la lettura dei testi un vero calvario. Magari non è più così e potrei rivedere questo giudizio, un giorno di questi ne prendo un numero e controllo.*

*Gigabyte invece la leggevo con piacere soprattutto in treno. Da sempre faccio il pendolare ho questa fortuna/scalogna di dovermi servire dei mezzi pubblici. Dato che per scelta e per comodità non prendo quasi mai la macchina, dispongo di una certa quantità di tempo da dedicare alla lettura.*

*Il sottotitolo della rivista: "CD-ROM, Shareware-PD, Telematica & ALTRO" testimonia la "mission" della rivista.*

*Il numero di novembre '94 si presenta con una copertina non particolarmente bella ma comunque molto ordinata dalla quale si evincono facilmente gli argomenti "speciali" che la redazione vuole mettere in risalto. Scopriamo così che possiamo aspettarci uno "Speciale Emulatori", un articolo "Source code" sui programmi che permettono di costruire facilmente games e una panoramica delle tecniche di cattura dello schermo mentre sono in funzione altri programmi: lo "Screen Grabber".*

*A pagina 10 viene spiegato il servizio "Pegaso" che altro non è che una classica BBS ma sul 144. Un modo per farsi pagare l'accesso insomma! Da come ne parlano, giustificandosi ogni tre righe, non credo che sia stata una iniziativa di successo e quantomeno la redazione deve essere stata subissata di proteste. D'altronde, complice la SIP, il 144 era identificato come*



| | LITIL DIVIL | THEATRE OF DEATH PSYGNOSIS | PACIFIC STRIKE ORIGIN | HEXX PSYGNOSIS | SPEAR OF DESTINY ID SOFTWARE |
|---|---|---|---|---|---|
| PCXDUMP | | | | | |
| SCR2GIF | | | | | |
| PCXCAP | | | | | |
| CAPTURE | | | | | |
| VGA2EPS | | | | | |
| VGACAP | | | | | |
| VGAGRAB | | | | | |
| GRABBER | | | | | |
| ST | | | | | |

*servizio erotico. Vi ricordate le polemiche sull'abilitazione automatica dell'acesso? Le associazioni dei consumatori vinsero quella battaglia e l'accesso venne chiuso ma il tutto, pornografia compresa, si è trasferito pari pari, sul 166 che è abilitato di default. Una classica storia di italico disservizio da parte delle aziende a partecipazione e protezione statale e dei loro dirigenti e presidenti pagati a peso d'oro (preso dalle nostre tasche naturalmente)!*

*Molto bello l'articolo sugli emulatori, cioè quei programmi che su PC ti illudono di avere sotto la tastiera un C64 o uno Spectrum. C'è anche l'inverso: il PC emulato su un Amiga. Curiosamente sembra che la molla principale che spinge a sviluppare ed adottare simili software è quella dell'emulazione dei giochi. Ho cercato di ricordare quale fosse il livello dei giochi su PC, ma sinceramente l'utilizzo ludico non è mai stato al primo posto fra i miei interessi informatici. Credo comunque che siano stati almeno allo stesso livello di quelli possibili via emulazione.*

*Non solo giochi comunque: quattro pagine sono dedicate alla recensione del cd-rom "Il dizionario della lingua italiana Devoto-Oli".*

*Tecnicamente perfetto l'articolo sulla cattura dello schermo. il cosidetto "Screen Grabbing". Bisogna pensare che i programmi sono principalmente DOS e quindi non ci si può arrangiare con la clipboard di Windows. Il DOS è notoriamente un ambiente mono-task e pertanto vanno messi in atto una serie di trucchi per "fregare" i programmatori che "trappano" la routine di input della tastiera sostituendole con le proprie, soprattutto nei programmi "giochi".*

*Completo ed interessante l'articolo che descrive le funzionalità del progrmma "Page Plus 3.0". Si tratta di un completo programma per il DTP (Desk Top Pubishing) funzionante in Windows (versione 3) che si propone come alternativa ai programmi molto più costosi, Ventura Publisher in testa.*

*Buona parte della rivista è dedicata alla recensione di giochi. È evidente che la vocazione "home" della pubblicazione non possa ignorare quello che è il mercato di punta. Il PC viene usato a casa moltissimo per i giochi, anzi, a giudicare dai dati di vendita sembra che sia l'unico utilizzo "ufficiale" delle macchine comperate per casa.*

*È strano: si comprano i giochi e si copiano i programmi office! Perché? Evidentemente è una questione principalmente di prezzo ma non solo. I giochi possono permettersi degli schemi di protezione che altri non potrebbero mai addottare; si pensi ad esempio all'inserimento di codici diversi ad ogni lancio e che si trovano sparsi sulle pagine dei manuali. Se Microsoft avesse solo tentato una cosa simile si sarebbe auto-esclusa da qualsiasi possibilità di fare cassetta con il software da ufficio.*

*Curioso il software "Klik & Play", una sorta di laboratorio per la costruzione di giochi. Tramite un'interfaccia abbastanza intuitiva è possibile definire il movimento degli "sprite" sullo schermo, le loro collisioni, etc... fino ad avere un gioco "platform" perfettamente funzionante. Non sò che tipo di successo o diffusione abbia avuto un simile filone di software, certo che leggere le confessioni di chi il software shareware ha provato a produrlo ... .*

## Conclusione

*In definitiva un centinaio di pagine di piacevole lettura, scarse di pubblicità, con articoli abbastanza approfonditi. Un vero peccato, ripeto, che sia resistita solo qualche anno, ma evidentemente i tempi cambiano, le mode si evolgono e non rimane altro che pendere atto dei cambiamenti, ai quali tutti noi contribuiamo. Un appassionato di tecnica e di informatia in particolare, non può essere per sua natura un reazionario. Il mondo cammina, anzi corre!*

***[Sn]***

# Biblioteca

*Le monografie vecchie e nuove che rappresentano una preziosa risorsa per chi ama il mondo dei computer in generale.*

**Scheda**

Titolo: La Grande Storia Del Computer

Sottotitolo: Dall'Abaco All'Intelligenza Artificiale

Autore: Massimo Bozzo

Editore: Edizioni Dedalo (Bari)

Anno: 1996

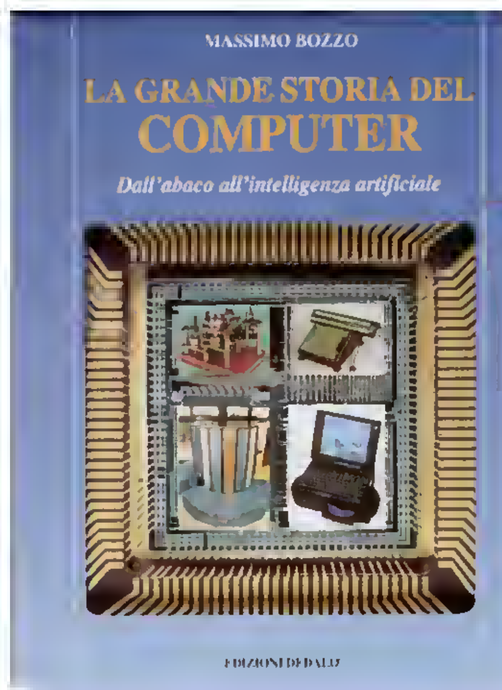
Lingua: Italiano

Pagine: 250

Prezzo: € 30,00 i.i.

ISBN: 978-88-220-4537-9

## La Grande Storia del Computer




Non mi trattengo dal dire che questa pubblicazione dovrebbe essere annoverata, almeno per quanto riguarda il nostro paese, tra quelle di riferimento. Non a caso, spesso la si trova citata nelle bibliografie di siti e pubblicazioni del settore di nostro interesse: il "retrocomputing".

Il libro tratta l'evoluzione delle macchine per il calcolo, le tecnologie e l'intelligenza artificiale seguendo un ordine rigorosamente cronologico. Secondo uno schema classico: suddivisione in capitoli nei quali viene evidenziato il fenomeno che caratterizza il periodo e, all'interno di questi, l'elenco di vicende e fatti rilevanti elencati in base all'anno. Per eventi maggiormente rilevanti sono presenti delle schede monografiche. Una organizzazione elementare, semplice, chiara lo rende simile ad una sorta di manuale e consultabile come tale.

Eppure, le vere qualità di questo libro, quelle cioè che lo rendono appunto "di riferimento", stanno nei contenuti e nel modo in cui vengono trattati. Queste qualità sono in parte esplicitate dall'autore stesso nella premessa: voler mantenere un equilibrio tra superficialità aneddotica ed eccesso di approfondimento scientifico; e non solo queste intenzioni sono abbondantemente mantenute, ma nonostante l'esposizione in forma cronologica, traspaiono pure: stile e una grande coerenza d'insieme. Sono sicuramente la passione ed il mestiere che permettono all'autore di rendere quasi avvincente l'esposizione degli eventi anche attraverso una struttura piuttosto asettica. Infatti, si possono individuare dei temi portanti che compaiono, scompaiono e riappaiono lungo tutto lo sviluppo del libro. Si tratta probabilmente di quegli argomenti di rilievo attraverso i quali

*è stata effettuata la cernita degli eventi da riportare. Però essi sono talmente ben distribuiti all'interno della trattazione e vengono così abilmente miscelati che costituiscono delle curiose trame delle quali si desidera, ad ogni pagina, conoscere l'evoluzione.*

*Ciò a cui si perviene, pertanto, è un'immagine di assoluto equilibrio e ponderatezza sia per il livello della trattazione sia per la varietà e modalità di trattazione degli argomenti. Dalle origini del calcolo (si parte dalla preistoria!), lo sviluppo scientifico, i sistemi di automazione, le macchine ma anche lo sviluppo delle diverse tecnologie correlate. Ancora: lo sviluppo industriale, la meccanografia, l'elettricità, l'elettronica. E poi la miniaturizzazione, le ricerche più avanzate, i tentativi falliti e quelli di enorme successo nell'industria oppure nel sapere. Insomma, oltre ad essere sviluppati con coerenza gli argomenti portanti sono pure tantissimi e tutti interessanti.*

*Come può succedere a molti lavori di alto livello essi, paradossalmente, possono sembrare perfettibili: si tratta soprattutto di un desiderio da parte del fruitore. Infatti, dopo essere stati coinvolti e soddisfatti leggendo un lavoro così ben fatto, il nostro impulso è quello di volere ancora di più, seppure senza sapere in che modo.*

*Per esempio, sarebbe piacevole un bel grafico cronologico curato con altrettanta coerenza e precisione (magari che confronti le capacità di calcolo delle diverse generazioni di macchine). Oppure, non saremmo mai abbastanza soddisfatti dell'egregio repertorio fotografico proprio per la sua varietà e originalità: così tanto originale che ci appare quasi carente solo perché ci lascia la voglia di poter averne ancora. Lo stesso vale anche per il numero di pagine, che pur essendo abbastanza abbondante (e comunque 250 sono coerenti con la tipologia di edizione) sembrerebbe ancora, quasi insufficiente!*

*Come è giusto osservare da parte di ogni bibliofilo che si rispetti, è necessaria anche un'ulteriore nota di merito rivolta alla qualità dell'edizione: pagine patinate, copertina cartonata ed un formato ampio e leggibilissimo ne giustificano il prezzo di copertina.*

*Considerando che il libro tratta aspetti di nicchia, non credo che gli esiti editoriali di una tale pubblicazione siano particolarmente gloriosi; e ciò è un peccato! Infatti questo testo riguarda lo sviluppo della scienza e della tecnologia in generale, ancor prima dell'informatica in particolare. E' pure un peccato che una così ben fatta trattazione di certi argomenti si sia dovuta fermare al 1996: sono passati ben dodici anni (tecnologicamente è tantissimo!) e sarebbe davvero bello sapere, nella stessa maniera e dalla stessa persona, come sono continuate tutte queste storie che ha voluto raccontarci.*

*Infine credo che per noi "retrocomputeristi" questo libro possa costituire una sorta di manuale di riferimento, se non altro per l'ampiezza delle vedute in cui inserisce il discorso relativo allo sviluppo tecnologico.*

***[Jb72]***

Apple Club

# Tutti i linguaggi dell'Apple (13)

*La mela come paradigma della programmazione*

```
Apple ][ CP/M
56K Ver. 2.20B
(C) 1980 Microsoft

A>dir
A: MUSIMP   COM : CONTINUE COM : TRACE    MUS : ARITH    MUS
A: ALGEBRA  ARI : EQN      ALG : ARRAY    ARI : MATRIX   ARR
A: SIGMA    ALG : LOG      ALG : TRGPOS   ALG : TRGNEG   ALG
A: DIF      ALG : INT      DIF : INTMORE  INT : TAYLOR   DIF
A: LIM      DIF : CLES1    ARI : CLES2    ARI
A>musimp

muSIMP-80  2.02  COPYRIGHT (C) 1980 MICROSOFT
LICENSED FROM THE SOFT WAREHOUSE

?
```

Boot del CP/M e lancio del programma muSimp. Come si vede il prompt è un punto interrogativo.

## Mumath/Musimp

Questa coppia di programmi, che si inserisce a buon diritto nell'insieme dei linguaggi di programmazione, sono progettati per la manipolazione simbolica delle espressioni matematiche.

Come diffusione dobbiamo dire che ne esistono più versioni, una delle quali sotto CP/M per cui è disponibile in tutte le piattaforme che ospitano un microprocessore compatibile (8080 e Z80) e quindi anche per Apple II con scheda Z80 di Microsoft o compatibile. Per preparare questo articolo abbiamo considerato proprio la versione sotto CP/M considerandola più "standard" in un certo modo e anche perché funziona con display a 80 colonne e si avvale di tutti quei programmi di supporto (ad esempio editor) che ne permettono l'uso più agevole.

A cosa serve un programma di calcolo simbolico? Serve! Prima di tutto fa la gioia di tutti i liceali alle prese con sviluppi in serie, studio di funzioni e riduzione di polinomi e poi fornisce un aiuto per tutti coloro che fanno della matematica un tool necessario alla loro professione.

Inutile ricordare che questi programmi (abbastanza grezzi per le limitate possibilità dei sistemi anni 80) hanno generato dei veri e propri mostri come Mathematica e Mathcad, tanto per citarne due dei più famosi. Questi sono programmi che permettono di esplorare le potenzialità della matematica in mo-

*niera del tutto impensabile fino ad allora.*

*Prima di tutto facciamo chiarezza su che differenze ci sono fra mu-Simpe muMath. La sorpresa è che in realtà muMath non esiste! O meglio: muMath è una "pachetizzazione" di funzioni create con muSimp e che mettono a disposizione un ambiente operativo adatto ad un particolare compito di calcolo. Ad esempio le radici di un polinomio, l'utilizzo dei numero immaginari, il calcolo matriciale, etc...*

*Poiché le capacità di memoria sia RAM che disco sono limitate in questi sistemi anni '80, si è pensato bene di non costruire un unico monolitico programma matematico, ma piuttosto fornire una serie di packages con i quali l'utente si prepara da solo la versione mu-Math (o le molte versioni) adatte ai suoi scopi di calcolo.*

*Il vero linguaggio di programmazione è quindi muSimp e noi, anche per ragioni di missione, ci soncentreremo su di esso rimandando l'utilizzo del calcolo simbolico via muMath ad un'altra occasione.*

*La più ecclatante differenza fra un linguaggio tradizionale e uno per la manaipolazione simbolica come mu-Simp, sta nel modo in cui vengono rappresentate e manipolate le variabili all'interno del linguaggio.*

*Prima di lanciarci nell'esplorazione delle possibilità del linguaggio, conviene che ci rendiamo conto di quali sono le possibilità che esso offre cariacando in memoria uno dei package e usandolo.*

*Sul dischetto, assieme all'interprete, troviamo vari file con diversa estensione. MuSimp assegna una estensione differente a tipi diversi di sorgenti. Ad esempio SYS è l'estensione che viene riservata al dump dell'intero ambiente, cioè al salvataggio completo di tutte le definizioni caricate in menoria e pronte per un successivo utilizzo.*

*ARI, MUS, DIF, ARR, INT, ALG sono altre estensioni utilizzate che hanno un preciso significato mnemonico: contengono le iniziali del package "padre" di quello salvato.*

*Lavorare con muSimp è un po' come giocare alle scatole cinesi: si definisce un package che abbisogna del solo muSimp (ed avrà estensione MUS), poi si specializza in un package "figlio" che prenderà l'estensione dalle iniziali del padre, e così via.*

*Figura 2.*
*Il comando di caricamento del file ARITH.MUS e il suo uso nel calcolo razionale.*

```
muSIMP-80  2.02  COPYRIGHT (C) 1980 MICROSOFT
LICENSED FROM THE SOFT WAREHOUSE

? RDS (ARITH, MUS);

@: ARITH


? 1/4 + 1/8;

@: 3 / 8

?
```

*Ad esempio fra la lista dei file scorgiamo ARITH.MUS il quale dipende da muSimp (MUS sono le iniziali del nome muSimp); a sua volta il package ALGEBRA.ARI è "figlio" di ARITH.MUS e così via.*

*E' chiaro che questa serie di dipendenze finisce per essere ingestibile o solamente difficile da ricordare; è per questo che ad un certo punto si fa un bel dump dell'ambiente e si mette tutto in un file "SYS" e buona notte!*

*Con muSimp caricato in memoria procediamo a caricare il package ARITH.MUS che contiene le funzioni per l'aritmetica (cioè numeri interi e numeri razionali).*

*Il comando è:*

```
RDS(ARITH, MUS);
```

*RDS è il comando di caricamento (probabilmente Read...qualche cosa), fra parentesi due parametri: il nome del file e la sua estensione, curiosamente definita a parte e non con la solita notazione NOME. EST.*

*muSimp termina i comandi con punto e virgola ";". Se non c'è il punto e virgola il comando non viene interpretato al momento di premere Return. Questo significa che si può scrivere tranquillamente una espressione su più righe andando a capo come se nulla fosse.*

*Il comando di caricamento impiega un certo tempo perché ogni definizione nel file da caricare viene interpretata e "accomodata" in memoria. Arith.mus è un file di testo, come ci si può facilmente rendere conto eseguendo un TYPE da CP/M.*

*Dopo il caricamento il sistema torna al prompt ("? " punto interrogativo più spazio) ed emette anche un bip, utile per richiamare l'attenzione qualora ci fossimo dedicati ad altro in attesa che l'operazione di lettura fosse terminata.*

*Arith è un package che permette la manipolazione simbolica dei numeri razionali (cioè quelli espressi da frazioni di interi).*

*Ad esempio:*

```
? 1/4 + 1/8;
@ 3/8
```

*La risposta, preceduta dal simbolo chiocciolina, come si vede non è un semplice numero reale ma rimane una frazione. Possiamo dire, con linguaggio da scuola media, che il sistema ha ridotto a fattor comune.*

*Qualcosa di più difficile:*

```
? (-24)^(1/3);
@ -2 * 3 ^ (1/3)
```

*Naturalmente l'operatore "^" è il simbolo dell'elevamento a potenza.*

```
? #E^(1/3) * #E^(2/3);
@ #E
```

*In muSimp esistono delle costanti predefinite. Una di queste è il numero di Nepero, base dei logaritmi naturali #E; un'altra è il Pi greco:*

```
? 123!
;

@:
1214630436702532967576624324188129585545421708848338231532891816182923589236216
7668831156960612640202170735835221294047782591091570411651472186029519906261646
73073390741981495296000000000000000000000000000

? █
```

#PI.

I calcoli numerici sono svolti immediatamente:

```
? 3 + 5;
@ 8

? 5! ;
@ 125
```

5! indica il fattoriale del numero, qualcosa che come sappiamo cresce molto ma molto velocemente al crescere del numero. Proviamo:

? 123! ;

Bhe, il risultato non lo trascriviamo: andate a vedervelo nella figura in alto. Direi che non c'è da scherzare!

Con questo ultimo calcolo abbiamo scoperto una cosa: muSimp non è limitato nel range dei numeri interi che rappresenta, essi possono essere lunghi a piacere. Ovviamente si dovranno fare i conti con la memoria e il tempo per calcolare queste quantità particolarmente grandi.

Se si vuole lavorare a livello algebrico, ad esempio alle rappresentazioni di polinomi, è necessario caricare il package ALGEBRA. ARI. Notare che l'estensione è ARI, le tre lettere iniziali del package che lo precede nella gerarchia, cioè ARITH.MUS.

Dopo il caricamento sono disponibili una serie di funzioni che permettono la riduzione dei polinomi, la ricerca delle radici, etc...

Un esempio di riduzione è visibile nella figura in fondo alla pagina.

Possiamo dare un'occhiatina all'interno del file ARITH.MUS per intuire qualcosa della sintassi del linguaggio. Essendo un file di testo basta caricarlo con un editor (ad esempio WordStar) o listarlo a video con il comando TYPE.

Ecco una semplice definizione di funzione:

```
FUNCTION SQUARE (EX1),
      EX1*EX1,
ENDFUN $
```

La funzione calcola il quadrato di un numero che viene passato come argomento (EX1). Il risultato è ov-

Chi ha detto che i numeri interi sull'Apple vanno da -38676 a + 38678?

La funzione di trasformazione frazionaria applicata ad un polinomio di secondo grado in tre variabili

```
? FCTR(6*X^2*Y -4*X*Y^2/2Z);
@: 2*Y*X*(-2*Y+3*X*Z) / Z

? █
```

viamente il numero moltiplicato per se stesso.

```
ENDFUN $
```

chiude la definizione di funzione.

Una funzione che stabilisce se due numeri sono uguali:

```
FUNCTION EQ (X, Y),
      WHEN INTEGER (X) AND
           INTEGER (Y),
           ZERO (X-Y) EXIT,
      FALSE,
ENDFUN;
```

La funzione esce con valore TRUE se X e Y sono due numeri e se sono uguali (la loro differenza è zero), oppure esce con FALSE in qualsiasi altro caso.

Come si vede nulla di stravolgente nel linguaggio muSimp rispetto ad altri idiomi ai quali possiamo essere abituati. Ad esempio che conosce il LISP troverà in muSimp più di una analogia.

Infatti anche nelle strutture interne muSimp assomiglia al LISP; magari non è così orientato alle liste ma ugualmente immagazzina i dati in una lista di puntatori che tengono traccia delle varie proprietà dell'oggetto. Ad esempio un puntatore (il primo della lista) punta al valore dell'oggetto, poi troviamo un puntatore al nome, alle altre proprietà e alla sua rappresentazione in caso di stampa.

Un dato in muSimp è classificato secondo tre tipologie: è un nome, oppure è un numero oppure è un nodo. L'entità NOME immagazzina tutti gli oggetti rappresentati da un nome simbolico all'interno del linguaggio; ad esempio variabili e funzioni. L'entità NUMERO è facile capire che si occupa di conservare in memoria le proprietà di un numero ed infine il NODO è in pratica una coppia puntata binaria utile a rappresentare strutture dati complesse in una forma ad albero binario.

Il NOME sottende in memoria una lista di quattro elementi:

NAME:

(Value, Property, Function, Pnames)

L'entità Pnames è la rappresentazione di stampa cui si accennava poco sopra.

Il NUMERO d'altro canto abbisogna di meno informazioni e la sua lista è:

NUMBER:

(Value, Sign, Vector)

In questo caso Vector è il primo puntatore ad una lista che contiene gli eventuali elementi di un vettore. In muSimp sono definibili array multidimensionali e funzioni che ne calcolano gli elementi.

Infine NODE:

(First, Rest)

ha una rappresentazione "tipo coppia puntata" molto simile al LISP. Infatti esiste una funzione built-in di muSimp che restituisce il primo elemento FIRST e una che restituisce il resto: REST.

*Imparare il linguaggio muSimp è tutt'altro che semplice e richiede una discreta dose di pazienza oltre che una conoscenza "ferma" delle basi matematiche. I risultati che si ottengono sono però all'altezza delle aspettative come testimoniano i numerosi esempi a complessità crescente che sono allegati nel package muMath. Si arriva fino allo sviluppo in serie e all'integrazione simbolica! Cose, come si vede, tutt'altro che banali.*

*Abbiamo accennato all'inizio dell'articolo alla presenza di una implementazione "nativa" per il 6502 e quindi proprio di natura "applistica".*

*I quattro dischetti che formano questa "distribuzione" (fra i quali uno vuoto non sono riuscito a capire perché), portano a bordo un sistema operativo diverso dal solito DOS Apple.*

*Pensavo inizialmente che fosse un caso ma poi ho trovato dei riferimenti Internet che citano il sistema operativo ADIOS-81 come modifica al nativo DOS e fatto apposta per distribuire muSimp.*

*Inizialmente non sapevo come usarli ma poi è bastato reperire un piccolo manualino e per quanto riguarda muSimp comportarsi esattamente come nella versione Z80, e le cose sono apparse chiare!*

*Nonostante la versione 6202 dia l'impressione di essere leggermente più veloce, tutto sommato la limitazione a 40 colonne del video fa propendere le mie preferenze per la versione CP/M, anche per la presenza di editor e utilities di gestione dei floppy più all'altezza della situazione.*

*Conclusione.*

*Abbiamo esplorato in questo articolo un linguaggio di programmazione alquanto strano. Un programmatore "normale" ne sarebbe probabilmente inorridito e chi veramente se lo godrebbe è invece il matematico "smanettone" e tutti quelli che fanno della curiosità una loro dote.*

***[Sm]***

```
 20 MUSIMP      COM ! 20 MUSIMPX    COM
  4 TRACE       MUS ! 17 ARITH      MUS
 11 ALGEBRA     ARI !  2 EQN        ALG
  4 SOLVE       EQN !  5 ARRAY      ARI
  7 MATRIX      ARR !  2 LOG        ALG
  3 TRGPOS      ALG !  4 TRGNEG     ALG
  3 DIF         ALG !  6 INT        DIF
  8 INTMORE     INT !  1 TAYLOR     DIF
  9 LIM         DIF !  4 SIGMA      ALG
  1 DISKCOPY    COM !

FREE ON 1:    1 K-BYTES

1 )MUSIMP
1 )RUN MUSIMP

MUSIMP-80 2.15 (03/01/82)
APPLE ][ ADIOS VERSION
COPYRIGHT (C) 1981 THE SOFT WAREHOUSE
LICENSED BY MICROSOFT, INC.

?
```

*muSimp in versione nativa per 6502 sotto ADIOS-81, una modifica al DOS 3.3 standard di Apple.*

# Retro Linguaggi

## LISP (parte 3)

*La storia dell'informatica è stata anche la storia dei linguaggi di programmazione.*



### Lisp e i processi condizionali

Affrontiamo in questa terza puntata l'aspetto della valutazione logica delle espressioni LISP.

Come in tutti i linguaggi di programmazione, anche il LISP implementa la logica booleana per valutare i valori di verità di una espressione.

Il valore vero viene esemplificato con la parola "true" o anche brevemente "T" (come al solito dipende dal dialetto che si usa). Il vaore opposto o falso (ma i puristi della logica matematica direbbero "non vero") con "false" o anche NIL, che corrisponde al puntatore vuoto.

Anche gli operatori di confronto vengono espressi in notazione LISP come liste:

(equal a b) -> true se 'a è uguale a 'b;

(greater a b) -> true se 'a è più grande di 'b, altrimenti NIL;

(number a) -> true se 'a è un numero;

(atom a) -> true se 'a è un atomo, NIL se 'a è una lista;

(null lista) -> true se lista è vuota o se l'argomento è NIL;

(zero a) -> true se 'a è zero.

Quelle sopra elencate sono le più comuni funzioni booleane presenti in LISP. Anche qui, in relazione ai dialetti in uso, gli operatori possono trovarsi espressi dai consueti simboli: '>' per maggiore, '=' per l'uguaglianza, etc...

Le espressioni booleane sono utilizzate nel flusso condizionale del programma. In LISP esiste la funzione COND che restituisce un certo valore in base ad una lista di possibilità selezionate tramite confronti.

Ad esempio definiamo la funzione max che restituisce il più grande fra due numeri:

```
(defun max (a b)
 (cond
      ((greater a b) a)
      (true b)
  )
)
```

Nella funzione COND in pratica si elencano le possibili scelte con una sitassi del tipo (condizione risultato). la valutazione avviene parten-

do dall'alto. Quindi ad esempio:

(max 3 2) -> 3

(max 2 3) -> 3

(max 3 3) -> 3

La sintassi (true b) indica una condizione sempre vera e quindi se la valutazione arriva fino a quel punto, il risultato della funzione è il parametro che segue il valore true.

Una piccola variazione sul tema della funzione max. Supponiamo di volere un risultato pari a 0 qualunque siano i due numeri passati come paramentri, se i due numeri sono uguali. La soluzione è che dobbiamo semplicemente aggiungere una condizione (in testa alle altre) che ci permetta di controllare questo caso speciale:

```
(defun max (a b)
 (cond
      ((equal a b ) 0)
      ((greater a b) a)
      (true b)))
```

Un'altro uso "classico" delle funzioni condizionali è quello che permette di trattare grandezze non definibili come numerosità a priori. Ad esempio si consideri il seguente problema: -"si scriva una funzione LISP che restituisca l'ultimo elemento di una lista data della quale non si conosce a priori la lunghezza".

Quello che dobbiamo fare per risolvere questo esercizio è iterare la funzione CDR fino a giungere ad una lista di un solo elemento, che sarà appunto l'ultimo elemento della lista sorgente.

```
(defun last (lista)
   (cond
      ((null (cdr lista)) (car
lista))
      ((last (cdr lista)))
   )
)
```

L'analisi è la seguente: nella prima condizione esaminata dalla COND si trova la cosidetta "condizione di uscita", soggetto tipico della programmazione ricorsiva. Se il CDR della lista è vuoto, allora l'ultimo elemento della lista è il suo CAR e siamo arrivati alla fine. Se non è così, per trovare l'ultimo elemento ri-applichiamo la funzione LAST ma ad una lista più piccola: quella senza il CAR.

Per la sua natura funzionale il LISP si presta molto bene ad implementare la ricorsione, cioè a quella tecnica di programmazione, ma anche di calcolo aritmetico, che consiste nell'applicare una funzione al risultato parziale della stessa per arrivare alla condizione elementare che può essere risolta in modo banale.

Vediamo un'altro esempio di utilizzo della funzione di selezione COND, questa volta applicata ad un problema di ricerca di un elemento in una lista originale. Cioè qualcosa per cui ad esempio:

(member 'a '(b c d a)) -> true

(member 'a '(b c d)) -> NIL

Si tratta di "andare a cercare l'elemento 'a all'interno della lista ed uscire alla prima occorrenza, oppure arrivati in fondo desistere con un bel NIL.

```
(defun member (a lista)
   (cond
      ((null lista) NIL)
      ((equal a (car lista))
true))
      ( true (member a (cdr
lista)))
   )
)
```

Anche nella definizione della funzione MEMBER abbiamo fatto uso della ricorsione, richiamando la funzione su se stessa ma ovviamente (regola fondamentale) su un insieme più piccolo.

Classico e immancabile l'esempio della definizione della funzione fattoriale. Come sapete certamente, ma lo ripetiamo per completezza, si definisce fattoriale del numero N il risultato dell'operazione:

Fattoriale(N) = N * Fattoriale(N - 1)

Come dire: Fattoriale(5) = 5 * 4 * 3 * 2 * 1

Con la condizione di limite: Fattoriale(0) = 1, altrimenti il fattoriale di qualsiasi numero sarebbe sempre zero!

La definizione in LISP è semplice:

```
(defun fattoriale (n)
   (cond
      ((zero n) 1)
      (true (fattoriale (sub n
1)))
   )
)
```

Attenzione: è molto facile sfondare lo stack, soprattutto su macchine retrò, quando ci si fa prendere la mano con la ricorsione!

Per terminare questa terza parte del corso facciamo una breve rassegna di altre funzioni che lavorano su stringhe e numeri e che implementano utili pezzetti di codice da usarsi eventualmente nei nostri programmi.

Calcolare la lunghezza di una lista.

```
(defun len(lista)
  (add L 1)
  (cond
  ((equal (car lista) NIL) 'L)
    (true (len (cdr lista)))
  )
)
```

Questa forma di calcolo della lunghezza di una lista non è proprio "pulita". Il problema è che abbiamo fatto due ipotesi che potrebbero anche non rivelarsi vere. La prima è che la variabile L non sia usata nell'environment, cioè non sia stato definito un atomo assegnando ad esso un valore; la seconda ipotesi è che la lista non deve essere vuota, altrimenti il risultato sarebbe comunque 1, il che ovviamente non è proprio il massimo della correttezza.

Dobbiamo disporre di "variabili loocali", come si direbbe in altri linguaggi e forse è più opportuno usare una implementazione iterativa per calcolare la lunghezza di una lista.

```
(defun len(lista)
   (prog (L X)
      (set L 0)
      (set X lista)
      LOOP
      (cond
         ((null X) (return L))
         (true (set L (add 1
L))
                  (set X (cdr
X))
            )
      )
   (go LOOP)))
```

*Abbiamo introdotto il goto, ebbene sì, LISP ne fa uso, al pari del BASIC non strutturato. Nella definizione prima di tutto la dichiarazione "prog" sta a significare che da quel punto parte un codice "locale" con due variabili L e X. La prima cosa che si fa entrando è instanziare queste due variabili rispettivamente a zero, per iniziare il contatore dio lunghezza, e alla lista sorgente. Poi X come lista sarà "decapitata" ad ogni iterazione.*

*Successivamente troviamo la parola isolata LOOP. Questo è un nome di una label e serve solo come riferimento per le istruzioni di salto.*

*La condizione successiva è l'algoritmo che calcola la lunghezza iterando la parte finale della lista fino a che la lista stessa (variabile X) non risulta vuota, cioè la condizione finale con restituzione del valore L (istruzione RETURN). Infine l'istruzione di salto che in LISP ha questa sintassi:*

*(GO <label>), dove <label> indica il nome del "bookmark" da dove riprendere l'esecuzione.*

*Nella prossima lezione faremo la conoscenza con altre importanti funzioni di base per la manipolazione delle stringhe e delle liste. In LISP è possibile trasformare una stringa in una lista di caratteri e viceversa... ma non anticipiamo troppo.*

*Alla prossima.* ***[Sm]***

# Emulazione

*I mondi virtuali a volte possono essere molto realistici...*

## ZX Spectrum on your PC

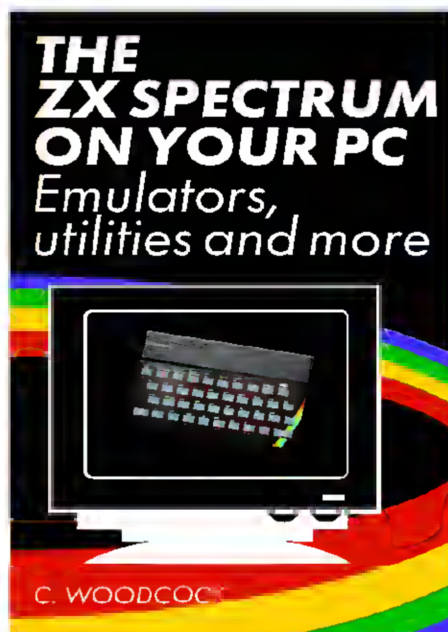



Figura 1.
La copertina della pubblicazione.

Occasione ghiottissima per gli ex possessori della macchina vintage forse più amata, lo Spectrum della Sinclair, per riavvicinarla attraverso l'emulazione sui moderni PC da scrivania.

Può darsi che a molte persone, antichi e presumibilmente felici possessori del capolavoro di sir Clive Sinclair, non abbiano trovato occasione o semplicemente il tempo e la voglia di mettersi ad approfondire il discorso sull'emulazione. Sembra tutto così naturale per noi che pratichiamo l'arte (perché di vera e propria arte si tratta) dell'emulazione, l'installare le innumerivoli beta dei vari progetti alla ricerca di quella "emulazione perfetta" o come era moda dire qualche anno fa "della madre di tutte le emulazioni". Non va negato che un minimo di studio è richiesto quando si decide di adottare uno di questi programmi: la corrispondenza dei tasti, i frame per secondo da regolare, il suono sì e il suono no, e per finire le immagini di dischi e cassette.

Personalmente ho sempre trovato molto interessante qualsiasi tentativo di portare il comportamento di un sistema su un'altro di pari o maggiore potenza. E' un po' come una "procrazione assistita", se mi passate il termine. Ora poi con la potenza dei PC che abbiamo sotto le dita, mai come adesso inutilizzata, non provare l'emozione dell'emulazione suona quasi blasfemo!

Colin Woodcock deve pensarla alla stessa maniera dal momento che è l'editore e uno degli animatori del sito ZXF, una fanzine dedicata allo ZX Spectrum[1] e un sito prezioso per tutti gli appassionati

[1] http://zxf.magazine.googlepages.com/home
[2] http://www.lulu.com/product/libro-a-copertina-morbida/the-zx-spectrum-on-your-pc/5595757

*del computer con l'arcobaleno.*

*Mantenendo fede allla propria missione anche il volume è prelevabile liberamente dal sito, oppure si può ordinare in edizione stampata (per chi ha posto) dal sito del servizio LULU[2].*

*Il volume affronta in maniera organica il progetto di portare uno ZX Spectrum su un PC attraverso l'adozione di uno degli emulatori disponibili sul Web.*

*Gli argomenti partono dalle basi dell'emulazione, proseguono con l'argomento della reperibilità dei programmi (le famose immagini) per lanciarsi poi verso argomenti più sofisticati quali l'emulazione delle periferiche (stampante e microdrive la fanno da padroni).*

*Utile il riassunto sulle varie release e sulle macchine Spectrum-Like che sono uscite nei quasi dieci anni in cui il sistema è rimasto in voga e sull'utilizzo dello stesso e delle periferiche dal punto di vista dell'utente.*

*Non si parla di programmazione, non avrebbe senso in un contesto dove principalmente si emula per giocare, ma le possibilità ci sono ovviamente tutte.*

*Concludono le 82 pagine del volume la panoramica sui cloni di origine russa e sul TR-DOS, il sistema operativo che aggiunge notevoli capacità di gestione allo Spectrum, anche se per i puristi queste "aggiunte" andrebbero evitate.*

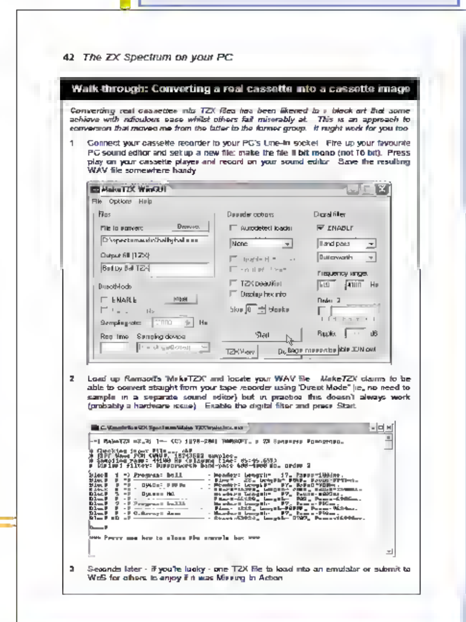
*La preferenza dell'autore va all'emulatore Spectaculator, ma le possibilità sono molte, comprese quelle che riguardano altre piattaforme (MAC e Linux).*

*Per finire l'autore ha pensato bene di accennare per completezza e anche per invitarne l'uso, al gruppo usenet dedicato alla macchina[3]*

*Conclusioni.*

*Questo testo mi è piaciuto parecchio: è curato, mai banale e si intuisce l'impegno che ne è stato profuso. Personalmente ritengo sia un volume destinato a durare a lungo e, magari con opportuni aggiornamenti, accompagnare i neofiti dell'emulazione o dello Spectrum alla scoperta di questo fantastico mondo virtuale e di questa fantastica macchina.*

***[L2]***



42 *The ZX Spectrum on your PC*

**Walk-through: Converting a real cassette into a cassette image**

*Converting real cassettes into TZX files has been likened to a black art that some achieve with ridiculous ease whilst others fail miserably at. This is an approach to conversion that moved me from the latter to the former group. It might work for you too*

1 Connect your cassette recorder to your PC's Line-In socket. Fire up your favourite PC sound editor and set up a new file: make the file 8 bit mono (not 16 bit). Press play on your cassette player and record on your sound editor. Save the resulting WAV file somewhere handy

2 Load up Ramsoft's 'MakeTZX' and locate your WAV file. MakeTZX claims to be able to convert straight from your tape recorder using 'Direct Mode' (ie, no need to sample in a separate sound editor) but in practice this doesn't always work (probably a hardware issue). Enable the digital filter and press Start.

3 Seconds later - if you're lucky - one TZX file to load into an emulator or submit to WoS for others to enjoy if it was Missing In Action

*Figura 2.*
*Una delle pagine ricche di informazioni riassume i passi necessari per convertire un file immagine in una cassetta audio reale.*

[3] comp.sys.sinclair

# TAMC

# Algoritmi di SORT (parte 6)

*Teoria e Applicazioni delle Macchine Calcolatrici: la matematica e l'informatica, le formule e gli algoritmi, la completezza e la computabilità, le strutture dati e tutto quello che sta alla base dell'informatica.*

## Counting Sort

*Finora gli algoritmi che abbiamo esaminato sono costruiti su un'unica ipotesi, cioè che il vettore da ordinare sia costituito da numeri interi senza altro vincolo.*

*Abbiamo annunciato che questa proiezione del problema realizza il principio di equivalenza algoritmica, cioè che se l'insieme da ordinare non è più un vettore di interi ma qualsiasi altra tipologia di dato, la complessità algoritmica rimane la stessa.*

*Per la verità, se ci pensiamo, dire che stiamo ordinando un insieme di interi su un computer ha di per se una limitazione: i valori sono compresi nel range rappresentabile dalla macchina nel particolare linguaggio che abbiamo deciso di adottare per la codifica. Questo non ci ha preoccupati, dal momento che teoricamente è possibile rappresentare numeri interi come sequenza, teoricamente infinita, di cifre. Cambierà il tempo di esecuzione, lo spazio di memoria, etc... ma non la complessità algoritmica.*

*Dove vogliamo arrivare con questa lunga premessa? Al fatto che se si conosce qualche caratteristica a priori dell'insieme da ordinare, allora si possono studiare algoritmi che traggano vantaggio da questa conoscenza intrinseca.*

*Un esempio è il Counting Sort, algoritmo di ordinamento che assomma due caratteristiche notevoli: è semplice da implementare e raggiunge una efficienza sorprendente: O(n) anche nel caso più sfortunato.*

*Counting sort è applicabile in un caso specifico e cioè con l'ipotesi aggiuntiva che i valori contenuti nell'insieme da ordinare siano paragonabili in grandezza alla dimensione N del vettore.*

*In altre parole: se N è il numero di elementi del vettore da ordinare e se tutti i valori sono inferiori ad una certa soglia S, dove S è all'incirca uguale a N, allora si cade nella ipotesi necessaria all'applicazione del Counting sort.*

*L'idea che sta alla base dell'algoritmo è la seguente: dal momento che sappiamo che un elemento p del vettore sta nel range di valori [0..S], ci saranno più o meno p elementi minori di p e quindi una buona posizione di partenza per p nel vettore ordinato sarà la posizione (p+1).*

*E' chiaro che questo primo passaggio non può essere sufficiente a restituire un vettore ordinato, a meno che il set di valori non sia proprio il sottoinsieme dei naturali [0...N].*

*Servono altri passaggi e anche due aree di appoggio, quindi molta memoria, ma ogni "giro" di raffinamento è lineare e pertanto conserva la complessità di ordine N, senza esplodere al crescere del numero di elementi.*

*L'informazione, che possiamo definire di cardinalità, dell'insieme di partenza, si sfrutta estendendola ai sottoinsiemi dello stesso. Cioè si fa il ragionamento seguente: se p è un numero appartenente al vettore da ordinare e abbiamo ipotizzato che la sua posizione finale sia più o meno all'indice (p+1), allora possiamo costruire un vettore che contenga per ogni indice il numero di elementi che lo precedono nella posizione. La cardinalità viene sfruttata per costruire un'area di appoggio all'ordinamento che sia a sua volta ordinata, cioè si lavora su una meta-conoscenza piuttosto che sulla conoscenza vera che sarebbero i reali valori nell'insieme.*

*Se si aggiungono vincoli di conoscenza del vettore di partenza, ad esempio se si suppone che tutti i valori sono diversi e sono nel range 1..N, allora è chiaro che l'ordinamento si riduce a spostare nella giusta posizione del vettore ordinato l'elemento il cui valore è anche l'indice stesso.*

```
Counting Sort

3 5 7 1 4 6 2
================
B = 0 2 0 0 0 0 0 | C = 1 2 3 4 5 6 7
B = 0 2 0 0 0 6 0 | C = 1 1 3 4 5 6 7
B = 0 2 0 4 0 6 0 | C = 1 1 3 4 5 5 7
B = 1 2 0 4 0 6 0 | C = 1 1 3 3 5 5 7
B = 1 2 0 4 0 6 7 | C = 0 1 3 3 5 5 7
B = 1 2 0 4 5 6 7 | C = 0 1 3 3 5 5 6
B = 1 2 3 4 5 6 7 | C = 0 1 3 3 4 5 6
================
1 2 3 4 5 6 7
```

*Figura 1. L'andamento di formazione del vettore ordinato B e del vettore "counting" C.*

*L'esempio che abbiamo usato nelle ultime lezioni (Mergesort e Heapsort) aveva proprio questa caratteristica, essendo un insieme dei primi sette numeri naturali senza ripetizioni:*

*[3, 5, 7, 1, 4, 6, 2]*

*Il vettore ordinato è ovviamente:*

*[1, 2, 3, 4, 5, 6, 7]*

*Per ottenerlo basta semplicemente prendere un elemento del vettore di partenza, ad esempio il primo, e lo si sposta nella posizione corrispondente al suo stesso valore:*

*[0, 0, 3, 0, 0, 0, 0]*

*e così via.*

*L'idea del Counting Sort è proprio la stessa, solo che è necessario un vettore di conteggio per rende-*

```
Program sortcount;

var
A, B   : array[1..7] of integer;
C      : array[0..7] of integer;
i, k, n : integer;


begin
  A[1] := 3;
  A[2] := 5;
  A[3] := 7;
  A[4] := 1;
  A[5] := 4;
  A[6] := 6;
  A[7] := 2;

  n := 7;
  k := 7;

  writeln('Counting Sort');
  writeln;
  writeln;
  writeln(A[1], ' ', A[2], ' ', A[3], ' ', A[4], ' ', A[5], ' ', A[6], ' ', A[7]);
  writeln ('================');

  { primo ciclo }
  for i:= 0 to k do C[i] := 0;

  { secondo ciclo }
  for i:= 1 to n do C[A[i]] := A[i];

  { terzo ciclo }
  for i := n downto 1 do begin
    B[C[A[i]]] := A[i];
    write('B = ', B[1], ' ', B[2], ' ', B[3], ' ', B[4], ' ', B[5], ' ', B[6], ' ', B[7],
' | ');
    writeln('C = ', C[1], ' ', C[2], ' ', C[3], ' ', C[4], ' ', C[5], ' ', C[6], ' ',
C[7]);

    C[A[i]] := C[A[i]] - 1;
    end;

  { quarto ciclo }
  for i := 1 to n do A[i] := B[i];

  writeln ('================');
  writeln(A[1], ' ', A[2], ' ', A[3], ' ', A[4], ' ', A[5], ' ', A[6], ' ', A[7]);
  readln;

end.
```

*re l'algoritmo più flessibile, dato che non accettiamo la banalità dell'esempio analizzato.*

*Nel listato riportato in questa pagina abbiamo la codifica nel solito Borland Pascal 7.0 sotto DOS (perché scomodare Java? :-).*

*Il cuore vero dell'algoritmo è il loop che costruisce il vettore di conteggio. Per adattarlo a tutti i casi, compreso quello della duplicazione di valori all'interno del set da ordinare, dovremmi aggiungere un ulteriore ciclo ai quattro presenti. Abbiamo deciso di non spingerci a tanto per non sporcare la percezione della semplicità di questa implementazione e perché ci interessava soprattutto ribadire che quando si affronta un problema di ordinamento o anche un problema informatico in generale, la conoscenza dei vincoli e delle condizioni al contorno determina a volte una semplificazione drastica dell'algoritmo e di conseguenza una maggiore efficienza e velocità di codifica.*

## Bucket Sort

*Dall'algoritmo Counting sort abbiamo imparato che qualora l'insieme dei valori da ordinare sia "non troppo cattivo", allora si possono fare semplificazioni notevoli. facciamo il seguente ragionamento: supponiamo che un insieme di N elementi appartenenti ai numeri naturali tali che l'i-esimo elemento sia tale che ogni valore sia compreso nel range $1<n_i<T$. Facciamo inoltre l'ipotesi che all'interno del sistema da ordinare gli elementi siano uniformemente distribuiti, cioè che si possa dividere l'intervanno in k "cestini" inserendo gli elementi che sono limitati nel range previsto per ogni bucket. L'uniforme distribuzione degli elementi assicura che nei k*

*bucket finiranno pochi elementi.*

*A questo punto si possono isolare questi sottoinsiemi e procedere al loro ordinamento utilizzando un algoritmo efficiente su insiemi piccoli e successivamente procedere con il "merge" di questi sacchettini di valori che risultano già a posto dal punto di vista dell'operazione di ordinamento.*

*Si tratta quindi di un algoritmo che prende vantaggio dalla preesistente ipotesi di riuscire a costruire dei sottoinsiemi trattabili con poca dispersione di risorse.*

*La predisposizione dei bucket e il loro riempimento è evidentemente una operazione lineare, cioè di ordine N, così come l'algoritmo di ordinamento che si applicherà ai singoli bucket, come si avrà l'accortezza di scegliere. Risulta infine che il Bucket Sort ha O($n^2$) anche nel caso più sfavorevole.*

*Riportiamo l'algoritmo in una pseudo-codifica:*

*BUCKETSORT( A, n, s , t )*

*Sia l'insieme A [1..n] il vettore da ordinare, s e t i limiti inferiore e superiore della distribuzione dei valori degli elementi di A.*

*Si divida il range 1..N in k elementi B (bucket) k = (s-t)/n*

*Per ogni elemento di A inserirlo nel Bucket giusto $B_i$.*

*Per ogni Bucket si proceda ad un ordinamento utilizzando ad esempio un InsertSort.*

*Sostituire gli elementi di A con la sequenza ricavata dai k Bucket:*

$A_j = B_{ki}$

## Conclusione

*Siamo così giunti alla conclusione del nostro mini viaggio fra i principali algoritmi di ordinamento che si utilizzano normalmente nell'ambito applicativo.*

*Così come abbiamo mostrato in questa puntata conclusiva, per procedere con qualcosa di diverso dai classici ormai consolidati, è necessario fare delle ipotesi sul vettore di elementi da ordinare. Sono necessarie cioè delle meta-conoscenze per attivare la strategia migliore nei confronti del problema che si intende affrontare.*

*Questa situazione è comune nell'ambito dell'informatica e chi ci lavora o intende farne la propria professione, ne incontrerà parecchie di queste situazioni.*

*Buon proseguimento a tutti.*

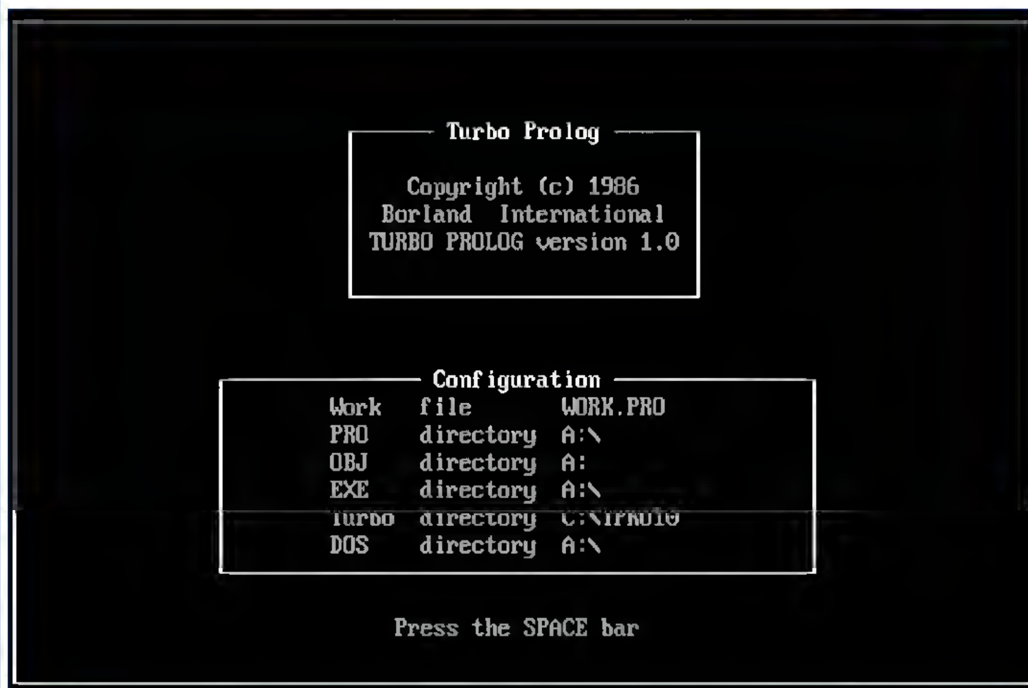
***[Sm]***

# Retro Software



*Non solo l'hardware ha segnato la storia dell'informatica personale, ma e soprattutto il software, il motore dei tanti sistemi che ci hanno accompagnato alla scoperta di questo fantastico mondo.*

## Borland Turbo Prolog 1.0




*Minimalista lo splash screen al primo lancio dell'ambiente. Ci ricorda che dobbiamo settare le variabili di environment.*

*Oggi vogliamo cimentarci con un linguaggio di programmazione che ha visto un periodo di inaspettata popolarità attorno alla metà degli anni '80 ma che poi è rapidamente caduto nel dimenticatoio.*

*Il grande fermento nell'evoluzione dei linguaggi di programmazione seguito dalla disponibilità diffusa delle piattaforme personal per il calcolo automatico, generò tutta una serie di tentativi ed errori tesi principalmente al superamento delle idiosincrasie insite nella sintassi del BASIC, linguaggio di elezione delle piattaforme home e personal delle prime generazioni.*

*L'impulso dato a questo comparto dalla comparsa del pc a basso prezzo è stato altissimo, complice anche la potenza elaborativa che andava crescendo di generazione in generazione e dall'approcciarsi alla teoria da parte di moltissimi ricercatori a livello mondiale.*

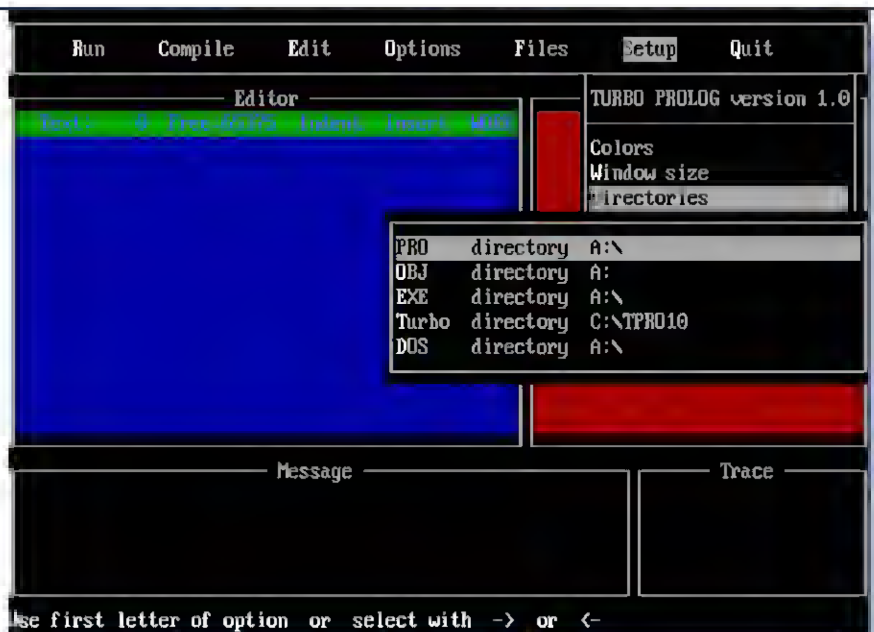
*Prolog sembrò avere quel tocco di magia che permetteva di slegare la descrizione di un problema dalla sua codifica a livello di linguaggio formale. Cioè: abbiamo un problema, lo descriviamo e lasciamo che sia la macchina a trovare la strada per risolverlo. Affascinante e ovviamente incline all'idea di una "Intelligenza Artificiale" nascosta da qualche parte nei chip e che andava semplicemente svelata, mentre nessuno dubitava della sua esistenza. Una sorta di fede*

*quindi, una convinzione che bastava trovare l'approccio giusto e improvvisamente quel mucchio di ferraglia si sarebbe trasformato come d'incanto in una macchina intelligente!*

*Anche la Borland, conosciutissima ed apprezzata software house produttrice dei compilatori "Turbo", si fece attirare da un mercato di nicchia ma con buone prospettive di crescita e fece uscire dal suo cappello a cilindro una implementazione del Prolog che a buona ragione venne appellato con l'aggettivo di "Turbo".*

*E in effetti il prodotto rimane fedele alla filosofia aziendale distaccando anni luce la concorrenza sul fronte della velocità di esecuzione. Borland poi fece tesoro della sua esperienza negli environment di programmazione corredando anche il Prolog di quelle features integrate che tanto servizio stavano rendendo al Pascal e al C dal punto di vista degli strumenti di sviluppo.*

*Per raggiungere l'efficienza il Turbo Prolog sacrifica qualche cosa allo standard: prima di tutto si "dialetizza", prende cioè una strada diversa per la sintassi rispetto al Prolog "ufficiale" che era quello dell'università di Edinburgo, poi chiamato "Edinburgh Prolog". La limitazione più importante non è comunque questa, ma piuttosto il pegno che Turbo Prolog paga per effetto della compilazione alla valutazione delle clausole dinamiche (vi spiego dopo cos'è una clausola).*

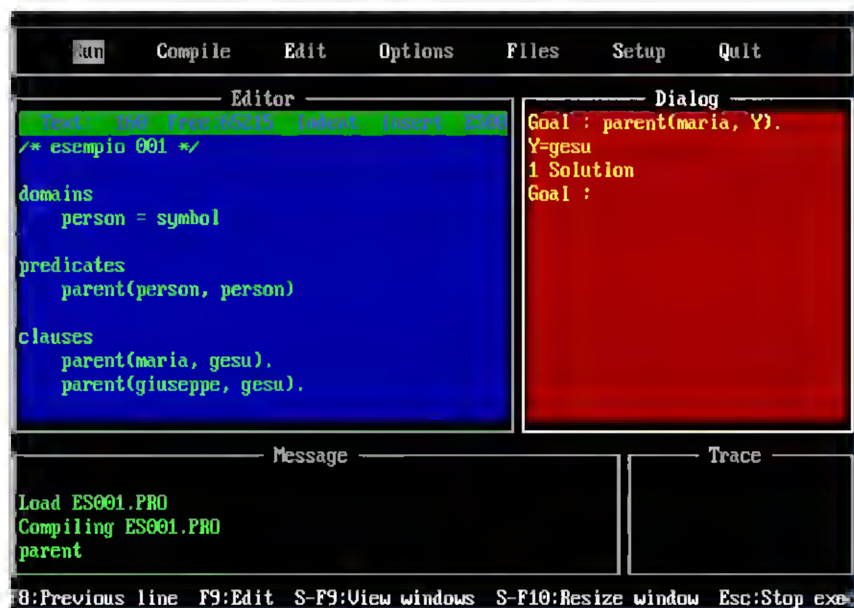



*Prima operazione: settare le directory.*

## Il prodotto

*Non è difficile procurarsi la verione 1.0 o la successiva 1.1 in rete. Non sono sicurissimo che sia stato rilasciato free da Borland, ma sicuramente per uso personale nessuno verrà mai a contestarvi di averne una copia in casa! Fra l'altro un'altra azienda, la Prolog Development Center (PDC) ne ha rilevato i diritti di sviluppo e produce un Visual Prolog (vedi bibliografia per i riferimenti), con anche una versione "personal" priva di licenza.*

*Turbo Prolog 1.0 viene rilasciato su un'unico Floppy da 5,25" ad alta densità e occupa circa 400 Kb una volta installato su hard disk. L'ambiente funziona anche su un sistema a due floppy, nel cui caso il file README consiglia come dividere i file fra i due supporti in modo da avere in linea il compilatore e contemporaneamente poter disporre di spazio su disco per lo sviluppo dei programmi.*




*Esiste l'applicativo INSTALL.EXE ma è anche sufficiente copiare i file in una directory del disco C: e poi eseguire l'applicativo PROLOG.EXE (di circa 200 KB) per trovarsi nell'ambiente di sviluppo.*

## Uso

*La prima cosa da fare è passare per il Setup e configurare le directory ed eventualmente i colori delle varie finestre che formano l'environment. Il funzionamento è il seguente: nella finestra Editor (raggiungibile dalla voce "Edit" sul menù) abbiamo il file in sviluppo che di default si chiama "WORK.PRO". La finestra Dialog è la parte interattiva, dove l'utente inserisce le domande che pone al sistema; poi abbiamo una window per i messaggi e una di trace.*

*Il mouse non è supportato per cui ci si deve arrangiare con la tastiera e gli eventuali tasti funzione. L'interazione con l'ambiente ne risulta un po' appesantita dal continuo ricorso al tasto Escape, ma siamo comunque in linea con quello al quale eravamo abituati anche con altri prodotti. Anche la dimensione utile per le finestre non è proprio il massimo, comunque le finestre dei messaggi e di trace si possono chiudere per avere maggiore spazio per l'editor e l'esecuzione.*

*La compilazione può avvenire in memoria oppure produrre un file oggetto (estensione OBJ) o direttamente un eseguibile (estensione EXE). Produrre un eseguibile comporta aver progettato un programma che non si aspetta una interattività utente, mentre l'uso più normale di un programma Prolog è l'esplorazione delle risposte a fronte del db della conoscenza acquisito dai fatti e dalle regole.*

*Turbo Prolog è un compilatore, il che significa che si predispone un sorgente, lo si compila e lo si manda in esecuzione. Un programma Prolog però non è come un classico programma che è in esecuzione dalla prima istruzione attraverso il flusso predisposto dal programmatore. Qui si tratta di compilare una serie di fatti e di regole, che poi sono la "conoscenza" del sistema in un certo contesto, poi in maniera interattiva si pongono delle domande per avere le conseguenti risposte.*

*Nelle figure che corredano l'articolo un semplice esempio. Abbiamo tre sezioni:*

*"domains" dove si dichiarano le variabili*

*"predicates" che contiene le regole di deduzione*

*"clauses" che è il database delle conoscenze.*

*In questo semplice esempio ispirato al Natale, abbiamo la dichiarazione di "person" come variabile (symbol significa che assumerà un valore nel programma); la regola è che per essere parent bisogna che ci siano due persone, evidentemente; infine le clausole che stabiliscono dei fatti. Qui vediamo che gesù (le costanti in Prolog si dichiarano con le lettere minuscole, mentre le maiuscole sono riservate alle variabili) ha due genitori: maria e giuseppe (senza sottilizzare troppo :-))*

*Dicevamo che Prolog da delle risposte e lo fa agendo sui predicati; possiamo chiedere ad esempio:*

*parent(X, gesu)*

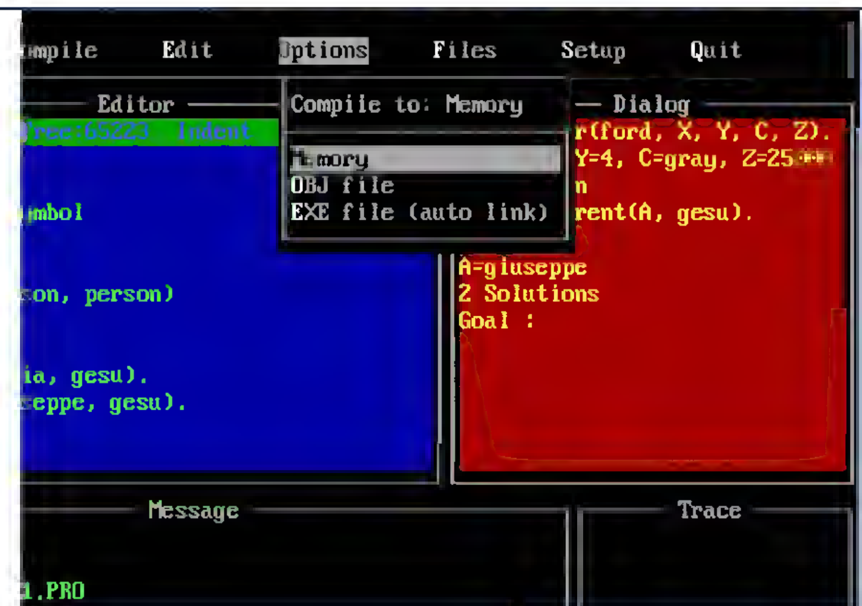
*per avere due soluzioni per la variabile run-time X: maria e giuseppe.*

*Peraltro si potrebbe anche chiedere:*

*parent(maria, Y)*

*per avere come risposta gesu.*

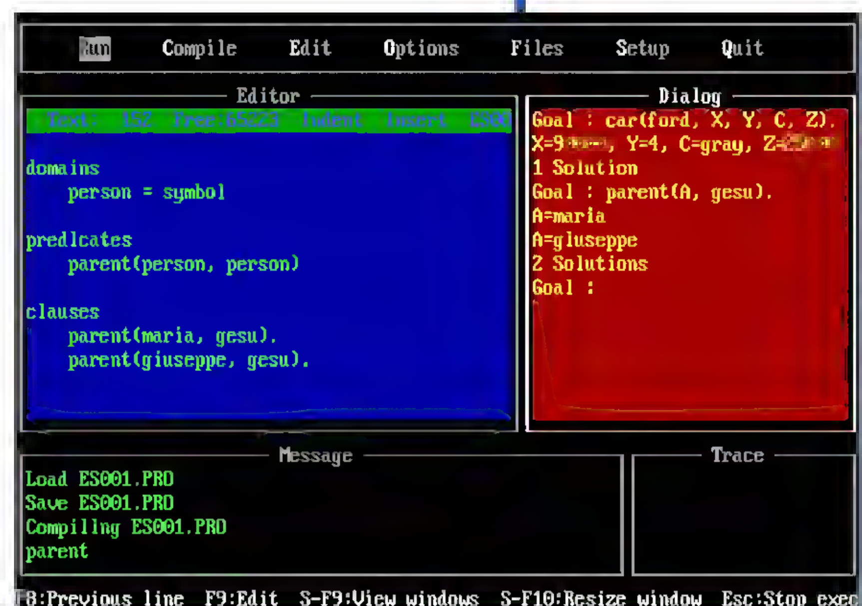
*Queste domande che vengono rivolte al sistema si chiamano in gergo "goal" che sta ad indicare l'idea che il sistema cerca di soddisfare il predicato basandosi sulle clausole ad esso conosciute e che possono*




*anche incrementarsi a run-time.*

*Il funzionamento di un sistema Prolog è basato su un principio teorico studiato dal logico John Alan Robinson nel 1965 che andava cercando un metodo matematico per rappresentare gli assiomi della logica dei predicati. Il suo principio dice in pratica che si può ridurre una clausola in forma normale, chiamata "forma di Horn" e che esiste un metodo meccanico in grado di dare delle risposte di verità (se esistono).*

*Non è il caso di approfondire*



```
/* Esempio numero 13 */

domains
   title,author = symbol
   pages        = integer
   publication  = book(title,pages)

predicates
   written_by(author,publication)
   long_novel(publication)

clauses
   written_by(fleming,book("DR NO",210)).
   written_by(melville,book("MOBY DICK",600)).
long_novel(book(Title,Length))
       :-written_by(_,book(Title,Length)), Length>300.
```

*questo aspetto teorico in questo momento, serve semplicemente ribadire come il Prolog sia un linguaggio nato con solidissime basi teoriche e che è in grado di rappresentare in forma inferenziale il concetto di conoscenza basata su fatti e regole.*

*Un altro sorgente tratto dall'esempio numero 13 presente sul dischetto è visibile nel box di questa pagina.*

*Si tratta di agire su un dominio tipico dell'editoria. Abbiamo dei libri caratterizzati dal nome dell'autore, dal titolo e dal numero di pagine. Si predispongono due predicati:*

*written_by che associa l'autore alla pubblicazione; la pubblicazione è una costante formata dalla coppia (titolo, pagine).*

*Long_novel è il predicato che risponde alla domanda se esistono, e quali sono, i libri classificabili come novelle lunghe (romanzi).*

*Vediamo nella sezione clauses come è programmato il predicato long_novel:*

```
long_novel(book(Title,Length)):-
written_by(_,book(Title,Length)),
                    Length>300.
```

*Cioè, detto in parole: un romanzo è una coppia (titolo, pagine) di qualsiasi book la cui lunghezza è superiore a 300 pagine. Un criterio di classificazione brutale, ma non stiamo facendo un corso di biblioteconomia :-)*

*Interessante il corpo della funzione/predicato long_novel, con l'uso del predicato written_by che fa uso del "segnaposto "_" (underscore), che significa in pratica "non importa cosa ci sia qui".*



## Conclusione

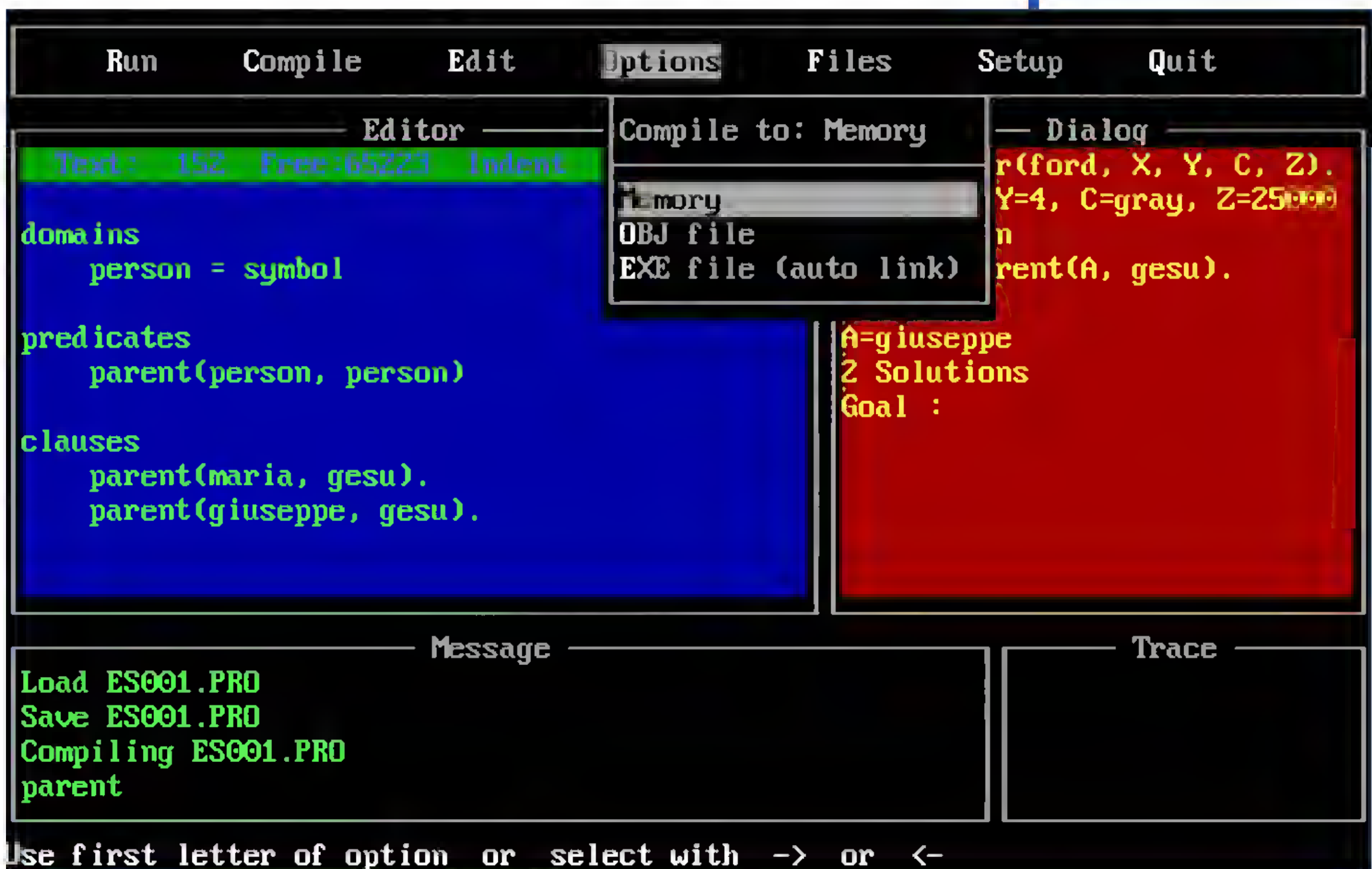
*Esaminando gli esempi a corredo del programma, ovviamente il manuale e qualche buon libro di fondamenti della programmazione in Prolog, fra l'altro abbondanti in Internet, è abbastanza facile impadronirsi dei primi rudimenti e soprattutto della sintassi del Turbo Prolog. Questa appare a prima vista inutilmente complicata rispetto al Prolog classico, ma evidentemente si tratta di prendere confidenza con il dialetto specifico. Molto meno facile è invece raggiungere un livello produttivo decente con un linguaggio di programmazione che si discosta e di molto dalla classica sequenza di istruzioni.*

*Possiamo dire che il Turbo Prolog è un prodotto della Borland che tiene fede alla fama dell'azienda ben più nota per il Turbo Pascal e successivamente per il Delphi sotto Windows. Se il Prolog avesse sfondato veramente nel mondo home e soprattutto professional, la Borland si sarebbe trovata indubbiamente in prima linea sul mercato.*

*Non è andata così, anche per l'obiettiva difficoltà di programmare scomponendo un problema in componenti logiche più che in intervalli di tempo, che è il paradigma classico dei linguaggi sequenziali.*

*Oggi il Prolog sopravvive nella programmazione logica appannaggio dei centri di ricerca nelle università, soprattutto inglesi (chissà come mai...), mentre è sparito del tutto dalla lista dei compilatori di utilizzo corrente sia del programmatore professionista che dell'hobbista appassionato dell'arte.*

***[Tn]***




*I tre possibili output della compilazione: memoria, oggetto o eseguibile.*

# L'intervista

Vari personaggi e amici incontrati qua e la per una chiacchierata sul mondo del retro computing.

## Intervista ad A. Zuech

*Intervista a Alessandro Zuech x Jurassic News*

*Alessandro lavora come IT manager in una società multinazionale che ha una sede nel nord Italia. Nonostante sia letteralmente immerso nelle nuove tecnologie, Alessandro ha saputo coltivare la passione per l'informatica delle origini dotandosi di una ricca collezione e soprattutto di una vasta cultura nel settore retro computing.*

*JN - Cominciamo con qualche informazione personale per farti conoscere meglio dai nostri lettori. Dunque ci dicevi che lavori in una società nel settore IT e hai iniziato a raccogliere materiale retrò più o meno attorno al 1990...*

*AZ - Lavoro in questo settore a partire dal 1989, ambiente in cui ho svolto per molti anni le attività come sistemista. Ricordo con particolare gioia le costanti e importanti novità che hanno contraddistinto tutti gli anni '90: le prime reti, i cambiamenti nell'hardware e nei sistemi operativi veramente 'epocali', la scoperta di internet e del mondo open source....*

*Tutte cose che hanno lasciato il segno e nei ricordi l'indelebile entusiasmo che contraddistigue qualsiasi attività quando si partecipa a realizzare qualcosa di unico.*

*JN - La solita domanda: perché e come ti è venuta l'idea, poi passione, di raccogliere i vecchi sistemi di calcolo?*

*AZ - Il mio rapporto con l'IT risulta antico, nonostante gli anni non siano ancora tanti. In un certo senso sono figlio d'arte, avendo mio padre lavorato in una grossa azienda del settore per più di trent'anni.*

*La passione di raccogliere sistemi di calcolo nasce molto tempo fa: ricordo quando da bambino accompagnavo qualche volta mio padre nei centri di calcolo. I video verdi, il rumore delle ventole di questi enormi elaboratori, il silenzio e l'attesa religiosa degli operatori quando attendevano in merito all'esito di qualche riparazione, come se si trattasse ogni volta di un'operazione chirurgica dall'esito incerto.*

*Costavano tantissimo, allora, quei grossi sistemi e davano relativamente poco rispetto ad ora..... ma, come oggi del resto, si viveva in un*

*contesto dove spesso veniva premiato il valore apparente ancor prima di quello reale.*

*E facevo domande, le domande che solo i bambini sanno fare.... una curiosità infinita che poi è continuata negli anni a venire.*

*Dopo essere entrato in possesso del mio primo 'vero' home, un vic-20 nel Natale 1982, un primo timore revenziale verso il tanto agognato computer si è via via trasformato in un vero hobby.*

*JN - Come mai una persona che s'interessa delle ultime tecnologie e probabimente per lavoro e svago possiede gli ultimi ritrovati in fatto di telefonini e computer mobility, si diverte a sporcarsi le mani con sistemi pesanti, lenti, limitati e spesso anche sporchi nel senso letterale dell'aggettivo?*

*AZ - Credo ci sia una forte componente di nostalgia per quello che tutti allora, senza internet abbiamo conosciuto. Le sgroppate a piedi attraverso la mia città per ottenere una rarissima rivista americana piena di listati, qualche nuovissima cassetta di giochi da copiare litigando innumerevoli volte con l'azimuth del registratore....*

*Un sacco di amicizie in più, di esperienze condite di emozione e toccate con mano.*

*Qualcosa di diverso e più profondo rispetto a Facebook e il WEB 2.0, ma ogni età ha le proprie espressioni.*

*Parallelamente mangiavo letteralmente le riviste con le prove delle nuove scatolette che arrivavano da oltreoceano, McMicrocomputer in particolare.*

*Mi è capitato di aver a che fare, recentemente, con qualche ex giornalista di quella testata per motivi professionali di tutt'altra natura: non vi dico l'emozione - neanche tanto repressa - quasi avessi a che fare con dei premi Nobel (con tutto il rispetto per Bo Arnklit, uno a cui il premio dovevano darlo sul serio).*

*Da qui l'interesse per l'hardware, cosa che mi ha portato a cambiare computer ogni 6-7 mesi. Comprati, smontati, provati, venduti tutti o quasi - almeno nella fascia di prezzo che uno squattrinato studente delle medie e superiori si poteva permettere: zx81, spectrum, c64, ti99, sega sc3000 (!!), msx, sinclair ql, amiga, ibm pc.....*

*JN - Parlaci di come hai iniziato o ricominciato a raccogliere vecchi calcolatori e soprattutto perché.*

*AZ - Appena entrato nel mondo del lavoro, finita di pagare l'immancabile automobile a rate, ho poi ricominciato a dar retta alla mia nostalgia, man mano che andavo avanti nello studio degli elaboratori moderni e man mano che realizzavo di conoscerne, nonostante gli sforzi, una parte sempre più infinitesimale.*

*Erano gli anni delle specializzazioni: l'esperto in hw faceva solo le riparazioni, l'esperto applicativo era un ragioniere convertito all'informatica con qualche rudimento di cobol e rpg, il sistemista parlava solo di reti e di sistemi operativi....*

*Non mi è mai piaciuto (e continua a non piacermi tutt'ora) riconoscermi in un solo comparto, senza visione d'insieme...*

*Se oggi studio il pezzo più vecchio della mia collezione, un sistema/3 IBM a schede perforate del 1969 da più di una tonnellata di peso, sono in grado di seguire il flusso dall'input dell'applicazione gestionale che 'gira' in un preciso momento fino al singolo nucleo di memoria a ferrite che viene elettricamente orientato per determinare la posizione 1 del singolo bit che viene modificato.*

*Si può fare in un sistema dove i singoli transistor sono visibili, come i cavi di interconnessione fra le varie schede, i sistemi operativi sono circa 2Kb di codice e i manuali riempiono*

*un'intera stanza documentando anche la composizione del metallo di cui è composto lo chassis....*

*Chiaramente, non che oggi sia realmente necessario a qualcuno tutto questo in dettaglio.... Quel che resta, da allora, è la visione fortemente 'sistemica' e concreta dei problemi da risolvere e la possibilità di operare su queste problematiche in team partendo da basi di conoscenza comuni e condivisibili.*

*Un qualcosa che forse oggi in questo settore si è perso e che si può anche riscoprire rileggendo il passato.*

*JN - parlaci della tua collezione, i pezzi più rari, quelli che ami in modo particolare...*

*AZ - La mia attuale collezione consta in alcuni sistemi IBM, fra cui il citato S/3, un series/1 del 1980, pc/xt e at oltre ad un as/400.*

*Lato microinformatica la parte del leone la fa la serie Sinclair che è completa, molti Commodore, apple II e mac, un sistema TI99 completato dalla rara peripheral expansion, atari, acorn - il tutto completato da molti libri, riviste e software di allora.*

*JN - Quali sono secondo il tuo parere le difficoltà più sfidanti che deve affrontare chi vuole dedicarsi a questo hobby?*

*AZ - Una delle operazioni più difficoltose e che richiedono più tempo è oggi quella del recupero del software da internet e della successiva registrazione su formati originali, floppies o cassette che fossero.*

*Un pc da collezione non è completo se non disponi dell'hardware funzionante e dal miglior sw di allora caricato da una 'ragliante' unità a dischetti.....*

*JN - Quali sbocchi pubblici può avere la tua passione: mostre, fiere, musei...?*

*AZ - Posso dire di aver scritto, non molto tempo fa, un testo finito su alcune riviste del settore riguardo delle novità prodotto su di un Apple IIe dotato di AppleWorks... una grande soddisfazione!*

*Mi fa molto piacere che in Italia siano sempre di più le iniziative legate al retrocomputing: iniziative quali JurassicNews ma anche tanti, tanti incontri di appassionati di questa e quella retroarchitettura che si incontrano ancora una volta 'davvero' e si scambiano pezzi, software e informazioni.*

*JN - Ci interessa conoscere il tuo punto di vista su alcune questioni "spinose" che animano il nostro mondo e in particolare se ritieni possa esistere un futuro per il nostro hobby. Un futuro "sostenibile" intendiamo, cioè una prospettiva di sviluppo che non sia la semplice collezione privata chiusa in un caveau.*

AZ - Credo che prima o poi potrebbe nascere qualcosa di strutturato anche da noi, magari legato all'iniziativa di qualche ente e/o università.

Del resto c'è stato un periodo, dove dalle parti di Pisa e Ivrea si inventavano cose che negli states neanche immaginavano!

Su questo tema attualmente gli USA sono al solito al primo posto. Sono tante le iniziative e i musei dedicati all'IT, soprattutto in California.

La capitale del retrocomputing risiede oggi senz'altro a Mountain View, nel cuore della Silicon Valley. Qui risiede il Computer History Museum (www.computerhistory.org) una iniziativa aperta dove è possibile trovare tutto ciò che conta, da alcuni pezzi dell'Eniac ad un PDP-1 funzionante, dalla serie completa di tutti i micro degli anni '80 fino ai box con i videogiochi arcade.

JN - Sappiamo che viaggi all'estero per lavoro. Approfitti di questa opportunità per entrare in contatto con appassionati di retro calcolo o visitare musei o collezioni o fiere, oppure in queste occasioni hai il tempo troppo limitato?

AZ - Parlando proprio del Computer History Museum, sono stato a visitarlo alcuni anni or sono e da allora sono diventato socio di questa iniziativa che farà in futuro parlare sempre più di se. Inutile dire che è un mio sogno nel cassetto trovare il tempo per trasferirmi da quelle parti un pò più a lungo del solito....

JN - Non ti chiediamo se ti piace Jurassic News, sarebbe puerile :-). Invece vorremmo chiederti in quale direzione migliorativa dovremmo lavorare per rendere JN ancora più interessante.

AZ - A mio avviso anche iniziative assolutamente encomiabili come quella di Jurassic News potrebbero estendere il proprio range di attività anche all'informatica pre-micro, alla fucina di idee e proposte rispetto a ciò che è stato e sicuramente... sarà.

**[Tn]**

## Sul prossimo numero in “edicola” a maggio:

- la prova del computer ORIC-1
- emulare sotto Linux
- retrocomputing: time machine
- storia dell’interfaccia grafica
- il racconto della serie Automatik
- la recensione del libro Il popolo del Joystick
- la quarta puntata del corso di LISP
- retroriviste: Computer World Weekly - Retrogames Times Monthly




... un numero speciale tutto d’oro!